

Cisco Secure Workload リリース 3.8 ユーザーガイド

初版：2023 年 5 月 19 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（ **www.cisco.com/jp/go/safety_warning/** ）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2023 Cisco Systems, Inc. All rights reserved.



目 次

第 1 章 はじめに 1

Cisco Secure Workload の概要 1

ワークロードについて 1

クイックスタートウィザード 2

セグメンテーションとマイクロセグメンテーションを使用する前に 3

マイクロセグメンテーション導入の一般的なプロセス 3

ベアメタルまたは仮想マシンで実行されるワークロードのマイクロセグメンテーションを設定する 6

クラウドベースのワークロードに対するマイクロセグメンテーションの設定 7

Kubernetes ベースのワークロードに対するマイクロセグメンテーションの設定 9

第 2 章 Cisco Smart Licensing 11

Cisco Secure Workload スマートライセンスの登録：CSSM ポータル 13

Cisco Secure Workload スマートライセンスの登録解除 15

ライセンス予約 16

特定ライセンス予約の更新 19

特定ライセンス予約の返却 19

Cisco Secure Workload スマートライセンスの登録：CSSM オンプレミス 20

スマートライセンスの同期 21

第 3 章 ソフトウェアエージェント 23

ソフトウェアエージェントの展開 24

サポートされているプラットフォームと要件 24

優れた可視性と適用のための Linux エージェントのインストール 25

Linux エージェントのインストールの要件と前提条件 25
Linux エージェントをインストールするためにサポートされている方法 25
Linux エージェントのインストールの確認 29
優れた可視性と適用のための Windows エージェントのインストール 30
Windows エージェントのインストールの要件と前提条件 30
Windows エージェントをインストールするためにサポートされている方法 30
Windows エージェントのインストールの確認 36
設定されたサービス ユーザー コンテキストの Windows エージェントの確認 37
サービスアカウントの変更 38
VDI インスタンスまたは VM テンプレートへのエージェントの展開（Windows） 38
Windows エージェントインストーラと Npcap：Windows 2008 R2 の場合 41
Windows エージェント フロー キャプチャ：Windows 2008 R2 を除くすべての Windows OS の場合 41
優れた可視性と適用のための AIX エージェントのインストール 42
AIX エージェントのインストールの要件と前提条件 42
エージェント スクリプト インストーラ方式を使用した AIX エージェントのインストール 43
AIX エージェントのインストールの確認 46
優れた可視性と適用のための Kubernetes または OpenShift エージェントのインストール 46
要件および前提条件 46
エージェント スクリプト インストーラ方式を使用した Kubernetes または OpenShift エージェントのインストール 48
優れた可視性のための Solaris エージェントのインストール 49
Solaris エージェントのインストールの要件と前提条件 49
エージェント スクリプト インストーラ方式を使用した Solaris エージェントのインストール 50
Solaris エージェントのインストールの確認 52
（手動インストールのみ）ユーザー構成ファイルの更新 52
その他のエージェント同様のツール 53
接続情報 54
セキュリティの除外 55
エージェントのサービスの管理 59

RHEL/CentOS/OracleLinux-6.x および Ubuntu-14 のサービス管理 59
RHEL/CentOS/OracleLinux-7.x 以降のサービスの管理 59
Windows サーバーまたは Windows VDI のサービス管理 60
AIX のサービスの管理 60
Kubernetes Agent インストールでのサービス管理 61
Solaris のサービスの管理 61
エージェントによるポリシーの適用 61
Linux プラットフォームでのエージェントの適用 62
Linux iptables/ip6tables 62
警告 63
WAF モードの Windows プラットフォームでのエージェントの適用 64
セキュリティが強化された Windows ファイアウォール 64
Secure Workload ルールと Windows ファイアウォール 64
セキュリティ プロファイル（Security Profiles） 65
効果的な設定および混合リストのポリシー 65
ステートフル適用 66
警告 66
WFP モードでの Windows プラットフォームにエージェントを適用 67
WFP（Windows フィルタリング プラットフォーム） 67
WAF における WFP の利点 67
エージェントによる WFP のサポート 67
エージェント WFP のサポートと Windows ファイアウォール 68
効果的な設定および混合リストのポリシー 68
ステートフル適用 69
構成された WFP フィルタの可視性 69
WFP モードでステルスモードフィルタを無効にする 69
設定済みの WFP フィルタの削除 70
既知の制限事項 70
Windows 属性のポリシーの設定 70
Windows OS ベースの推奨ポリシー設定 73
既知の制限事項 74

警告 74

Windows OS ベースのフィルタリング属性を使用したポリシーの確認とトラブルシューティング 74

AIX プラットフォームでのエージェントの適用 83

IPFilter 84

警告 85

既知の制限事項 85

エージェントのステータスと統計の確認 85

エージェント詳細の表示 86

ソフトウェアエージェントの設定 87

ソフトウェアエージェント設定の要件と前提条件 87

ユーザーのロールとエージェント構成へのアクセス 87

ソフトウェアエージェントの構成 89

エージェント設定プロファイルの作成 90

エージェント構成インテントの作成 99

エージェントのリモート VRF 設定の作成 100

インターフェイス構成インテントの作成 101

ワークロードプロファイルでの詳細なエージェントステータスの表示 103

エージェントのリホーム 105

リホームの有効化 105

リホームするエージェントの選択 107

リホームの無効化 108

適用が有効である場合のホスト IP アドレスの変更 108

ソフトウェアエージェントのアップグレード 109

UI からのエージェントのアップグレード 109

Kubernetes/Openshift エージェントの動作のアップグレード 113

ソフトウェアエージェントの削除 113

優れた可視性または適用 Linux エージェントの削除 113

優れた可視性/適用 Widnows エージェントの削除 114

優れた可視性または適用 AIX エージェントの削除 115

ユニバーサル Linux エージェントの削除 116

ユニバーサル Windows エージェントの削除 116
適用 Kubernetes または OpenShift エージェントの削除 117
優れた可視性 Solaris エージェントの削除 117
ワークロードエージェントにより収集されエクスポートされるデータ 117
登録 117
エージェントのアップグレード 118
構成サーバー 118
ネットワークフロー 119
マシン情報 119
エージェント統計情報 120
適用アラート 120
適用 UI アラートの詳細 125
適用アラートの詳細 125
適用アラートの alert_details の例 126
センサーアラート 126
センサー UI アラートの詳細 131
センサーアラートの詳細 131
センサーアラートの alert_details の例 132
ソフトウェアエージェントのトラブルシューティング 132
一般 132
エージェントの展開 132
Linux 132
Windows 134
Kubernetes 135
異常タイプ 137
非アクティブなエージェント 137
アップグレードの失敗 137
コンバート失敗 138
機能の変換 138
同期されていないポリシー 138
フローのエクスポート：Pcap オープン 139

フローエクスポート：HTTPS 接続 139
証明書の問題 140
Windows 140
NPCAP インストーラの証明書の問題 143
Windows ホストの名前変更 145
プラットフォームが現在サポートされているかどうかを確認する 146
Windows 146
Linux 146
AIX 146
Windows Installer の問題 146
必要な Windows サービス 147
Npcap の問題 147
Npcap がアップグレードされない（手動またはエージェント経由） 147
Npcap がインストールされない 148
Npcap が完全にインストールされているかどうかの確認 148
NPCAP のインストールまたはアップグレードの実行中のネットワーク接続の問題 150
Npcap との NIC チーミングの互換性の問題 150
ネットワークフローを報告しない VDI インスタンス VM 151
Npcap でのネットワークパフォーマンス 151
OS のパフォーマンスや安定性の問題 153
GPO の設定 153
クラスタ通信へのエージェント 154
接続のタイプ 154
接続状態の確認 155
SSL のトラブルシューティング 156
エージェント通信の概要 156
エージェント通信の IP トラフィックの構成 156
SSL/TLS 接続のトラブルシューティング 157
エージェントの操作 159

第 4 章 外部オーケストレータ 163

[外部オーケストレータ（External Orchestrators）] ページへの移動 165
外部オーケストレータの一覧表示 165
外部オーケストレータの作成 166
外部オーケストレータの編集 171
外部オーケストレータの削除 171
オーケストレータにより生成されるラベル 171
Secure Connector 172
技術的な詳細情報 173
Secure Connector の要件 173
Secure Connector クライアントの導入 174
プロキシ サポート 174
展開の概要 174
Secure Connector クライアントの導入 174
（オプション）特定のバージョンの Secure Connector クライアントを展開する 176
Secure Connector クライアントの状態の確認 178
Secure Connector クライアントステータス 178
Secure Connector クライアントのアップグレード 181
Secure Connector クライアントの削除 181
Amazon Web Services 181
前提条件 181
設定フィールド 181
ワークフロー 182
オーケストレータにより生成されるラベル 183
インスタンス固有のラベル 183
トラブルシューティング 183
Kubernetes/OpenShift 184
要件および前提条件 185
設定フィールド 185
オーケストレータのゴールデンルール 187
ワークフロー 188
Kubernetes ロールベース アクセス コントロール（RBAC）リソースに関する考慮事項 188

オーケストレータにより生成されるラベル 191
トラブルシューティング 191
VMware vCenter 192
前提条件 193
設定フィールド 193
ワークフロー 193
オーケストレータにより生成されるラベル 193
インスタンス固有のラベル 194
警告 194
トラブルシューティング 194
DNS 195
前提条件 195
設定フィールド 195
ワークフロー 195
生成されるラベル 196
警告 196
トラブルシューティング 196
DNS オーケストレータのフル/差分ポーリングの動作 197
サポートされない機能 197
Infoblox 197
前提条件 198
設定フィールド 198
ワークフロー 199
オーケストレータにより生成されるラベル 199
生成されるラベル 199
警告 200
トラブルシューティング 200
F5 BIG-IP 200
前提条件 201
設定フィールド 201
ワークフロー 202

オーケストレータにより生成されるラベル 203
生成されるラベル 203
F5 BIG-IP のポリシーの適用 203
F5 入力コントローラのポリシーの適用 205
警告 208
トラブルシューティング 208
Citrix Netscaler 208
前提条件 208
設定フィールド 209
ワークフロー 209
オーケストレータにより生成されるラベル 210
生成されるラベル 210
Citrix Netscaler のポリシーの適用 210
警告 212
トラブルシューティング 212
TAXII 212
前提条件 213
設定フィールド 213
ワークフロー 214
生成されるラベル 215
警告 215
トラブルシューティング 215
TAXII オーケストレータのフルポーリングの動作 215
Cisco Secure Firewall Management Center 215
オーケストレータにより生成されるラベル 216

第 5 章 コネクタ 217
コネクタとは 217
[コネクタ（Connectors）] ページへの移動 218
フローの取り込み用のコネクタ 218
NetFlow コネクタ 219

F5 コネクタ 226
NetScaler コネクタ 231
Cisco Secure Firewall コネクタ 235
Meraki コネクタ 241
ERSPAN コネクタ 246
エンドポイントのコネクタ 250
AnyConnect コネクタ 251
ISE コネクタ 256
インベントリ強化用のコネクタ 268
ServiceNow コネクタ 268
ServiceNow コネクタの設定方法 269
ServiceNow インスタンスの構成 270
ServiceNow レコードの処理 273
同期間隔の設定 273
ラベルを削除する Explore コマンド 274
テナントの VRF ID の検索 275
Explore コマンド UI へのアクセス 275
コマンドの実行 275
FAQ 276
制限 276
アラート通知用のコネクタ 276
Syslog Connector 277
Email Connector 280
Slack コネクタ 283
PagerDuty Connector 285
Kinesis コネクタ 286
Cloud Connector 288
AWS コネクタ 290
Azure コネクタ 308
GCP コネクタ 318
コネクタ用の仮想アプライアンス 327
仮想アプライアンスの種類 328

Cisco Secure Workload Ingest 328
Cisco Secure Workload Edge 330
仮想アプライアンスの展開 331
仮想アプライアンスのデコミッション 339
仮想アプライアンスのモニタリング 339
セキュリティに関する注意事項 340
コネクタのライフサイクル管理 340
コネクタの有効化 340
コネクタ関連情報の表示 343
コネクタの削除 344
コネクタのモニタリング 345
コネクタおよび仮想アプライアンスの構成管理 345
テストおよび適用 345
NTP の設定（NTP Configuration） 346
ログ設定 347
エンドポイントの設定 349
Slack 通知設定 350
PagerDuty 通知設定 350
Kinesis 通知設定 351
電子メール通知設定 351
Syslog 通知設定 352
Syslog のシビラティ（重大度）のマッピング設定 353
ISE インスタンスの設定 354
検出 354
LDAP 設定 355
削除（Remove） 361
トラブルシューティング 361
許可されている一連のコマンド 361
Show Logs 362
サービスログを表示 363
show running-config の出力結果 364

サービス実行設定の表示 365
システムコマンドの表示 366
Docker コマンドの表示 367
Docker インスタンスコマンドの表示 369
Supervisor コマンドの表示 371
スーパーバイザサービスのコマンドの表示 372
ネットワーク接続コマンド 373
ファイルの一覧表示 374
サービスファイルの一覧表示 375
パケットキャプチャ 377
コネクタのリスニングポートを更新する 378
アラート通知コネクタログ構成の更新 380
アプライアンスからのスナップショットの収集 381
コネクタからのスナップショットの収集 383
コントローラプロファイルの収集 384
コネクタプロファイルの収集 385
アプライアンスのコネクタアラート間隔のオーバーライド 387
コネクタのコネクタアラート間隔のオーバーライド 387
Hawkeye ダッシュボード 388
アプライアンスコントローラ ダッシュボード 388
サービスダッシュボード 389
AnyConnect サービスダッシュボード 390
アプライアンスとサービス DIO ダッシュボード 391
一般的なトラブルシューティングのガイドライン 392
ログ ファイル 394
コネクタアラート 395
アラート設定 395
アラートタイプ 395
アプライアンス/コネクタダウン 396
アプライアンスまたはコネクタシステムの使用率 397
コネクタ構成エラー 397

コネクタ UI アラートの詳細 398
アラート詳細 399
アラートの詳細の例 399
Cisco Secure Firewall Management Center 400
アラート設定 400
アラートタイプ 401
アプライアンス/コネクタダウン 401
アプライアンスまたはコネクタシステムの使用率 402
コネクタ構成エラー 403
コネクタ UI アラートの詳細 404
アラート詳細 404
アラートの詳細の例 405

第 6 章 インベントリ 407
ワークロードラベル 408
ラベルの重要性 408
サブネットベースのラベル継承 409
ラベルのプレフィックス 409
クラウドコネクタによって生成されたラベル 410
Kubernetes クラスタに関連するラベル 411
カスタムラベルのインポート 414
ラベルファイルのアップロードに関するガイドライン 414
ラベルキースキーマ 415
カスタムラベルのアップロード 416
検索ラベル 417
カスタムラベルの手動割り当てまたは編集 417
ラベルのダウンロード 418
ラベルの変更 418
ラベルの無効化 419
ラベルを削除する 419
ラベルの使用状況の表示 420

ラベルのメンテナンスプロセスの作成 421
範囲とインベントリ 421
スコープ 422
範囲フィルタ 424
フル範囲のクエリ 426
範囲へのアクセスの提供 427
範囲の表示 427
範囲を参照するフローの検索 428
新しい範囲の作成 429
範囲の重複 430
スコープの編集 431
スコープの削除 434
範囲ツリーのリセット 436
変更の確定 437
ログの変更 438
新しいテナントの作成 439
インベントリ 440
インベントリの検索 441
子範囲の提案 444
フィルタ 452
インベントリフィルタの作成 453
所有権の範囲に制限 454
範囲/フィルタ変更の影響を確認 455
範囲クエリ変更影響モーダル 456
メンバーシップの変更 456
依存関係 457
フィルタクエリ変更影響モーダル 458
メンバーシップの変更 459
依存関係 459
インベントリプロファイル 460
ワークロード プロファイル 462

ラベルと範囲タブ 462
[エージェントの正常性（Agent Health）] タブ 463
[プロセスリスト（Process List）] タブ 464
[プロセススナップショット（Process Snapshot）] タブ 465
[Interfaces] タブ 466
[ソフトウェアパッケージ（Software Packages）] タブ 466
[脆弱性（Vulnerabilities）] タブ 467
[エージェントの設定（Agent Configuration）] タブ 468
[エージェント統計情報（Agent Statistics）] タブ 469
[具体的なポリシー（Concrete Policies）] タブ 470
[コンテナポリシー（Container Policies）] タブ 472
[ネットワーク異常（Network Anomalies）] タブ 472
[ファイルハッシュ（File Hashes）] タブ 473
Software Packages 473
[Packages] タブ 473
Common Vulnerabilities and Exposures（CVE） 474
Windows パッケージと CVE 475
インベントリフィルタ 476
脆弱性データの可視化 476
ワークロード プロファイル ページ 477
[Packages] タブ 477
[プロセスリスト（Process List）] タブ 477
[プロセススナップショット（Process Snapshot）] タブ 478
[脆弱性（Vulnerabilities）] タブ 479
インベントリフィルタ 479
CVE ID ベースのフィルタ 479
CVSS（共通脆弱性評価システム）インパクトスコアベースのフィルタ 480
CVSSv2 ベースのフィルタ 481
CVSSv3 ベースのフィルタ 482
サービス プロファイル 483
ポッドプロファイル 484

近隣 485
近隣データの探索 486
地理データの探索 487
近隣の探索 496
近接アラート 502
アラートの設定方法 502
アラートの表示方法 504
アラート詳細 505

第 7 章 ポリシー 513
セグメンテーションポリシーの基本 513
ワークスペースを使用したポリシーの管理 514
ポリシーの操作：[ワークスペース（Workspaces）] ページへの移動 514
ワークスペースの作成 515
プライマリおよびセカンダリワークスペース 515
ワークスペース名の変更 516
範囲内のワークロードの表示 516
ワークスペース内の検索 516
ワークスペースの削除 519
ポリシーについて 521
ポリシー属性 521
ポリシーのランク：絶対、デフォルト、キャッチオール 522
ポリシーの継承と範囲ツリー 522
ポリシーでのコンシューマとプロバイダーについて 523
ポリシーの例 523
ポリシーの作成と検出 524
ポリシーの作成のベストプラクティス 524
ポリシーの手動作成 525
[ポリシーの追加（Add Policy）] ボタンが使用できない場合 527
特定の目的のためのポリシー 527
ネットワーク外部からのトラフィックをブロックするための InfoSec ポリシーの作成 527

差し迫った脅威に対処するポリシーの作成 528
脆弱なワークロードを検疫するポリシーの作成 528
ポリシーテンプレート 529
システム定義ポリシーテンプレート 529
カスタム ポリシー テンプレートの作成 530
テンプレートの適用 533
ポリシーの自動検出 534
ポリシー検出の詳細 535
ポリシーを自動的に検出する方法 536
1 つの範囲または範囲ツリーのブランチのポリシーの検出 538
ポリシー検出が適用されるワークロードの確認 541
ポリシーの自動検出 543
進行中の自動ポリシー検出の停止 545
自動ポリシー検出の高度な機能 545
ポリシーの承認 566
反復的なポリシーの変更 568
検出されたポリシーバージョンの表示、比較、および管理 570
ポリシー検出の Kubernetes のサポート 575
インポート/エクスポート 576
ワークスペースのエクスポート 576
Import 577
プラットフォーム固有のポリシー 578
Windows 579
Kubernetes と OpenShift 591
ワークロードのグループ化：クラスタとインベントリフィルタ 595
クラスタ 596
クラスタの信頼度 597
クラスタの表示 598
クラスタに変更を加える 599
クラスタをインベントリフィルタに変換する 601
クラスタの作成または削除 602

生成されたクラスタのバージョンの比較：差分ビュー 603
自動ポリシー検出の再実行中のクラスタ変更の防止 605
クラスタの承認 605
ポリシーの複雑さの対処 607
ポリシーの優先順位 608
ポリシーのグローバルな順序付けと競合の解決 608
ポリシーの順序と優先順位の検証 612
（上級）ポリシーの優先順位の変更 613
コンシューマとプロバイダーが異なる範囲にある場合：ポリシーオプション 615
（上級）クロス範囲ポリシーの作成 616
クロス範囲ポリシーのトラブルシューティング 628
有効なコンシューマまたは有効なプロバイダー 629
ポリシーの削除について 632
ポリシーの確認と分析 633
自動検出されたポリシーの確認 633
信頼度の低いポリシーへの対処 635
自動ポリシー検出結果のトラブルシューティング 636
ポリシーの視覚的表現 637
簡易分析 639
ライブポリシー分析 641
ライブポリシー分析の開始 642
ライブポリシー分析の停止 643
ポリシー分析結果：基本の理解 643
例：他の範囲で分析されたポリシーの影響 644
ポリシー分析の詳細 645
フローを調査するための推奨手順 647
過去のトラフィックに対して現在のポリシーをテストするポリシー実験の実行 649
ポリシーの変更後の、最新のポリシーの分析 651
ポリシーラベルのフラグ 651
分析されたポリシーバージョンの表示、比較、および管理 652
ポリシー分析のアクティビティログ 653

ポリシーの適用 654
エージェントの正常性と適用の準備状況の確認 654
ポリシーの適用の有効化 657
ポリシー適用ウィザード 661
コンテナへの適用 663
適用が予想どおりに機能することの確認 665
特定のワークロードの適用されたポリシー（具体的なポリシー）の表示 665
エージェントの適用が有効になっていることの確認 667
適用されたポリシーがエージェントにプッシュされていることの確認 667
エージェントのポリシーが多すぎる場合 668
適用されたポリシーの変更 669
新規および変更されたポリシーの適用 669
適用されたポリシーバージョンの表示、比較、および管理 669
適用されたポリシーを以前のバージョンに戻す 671
ポリシー適用の無効化 671
ポリシー更新の一時停止 672
適用履歴 673
ポリシーバージョン（v* および p*）について 673
ポリシーバージョンの比較：ポリシーの差分 676
アクティビティログとバージョン履歴 678
古いポリシーバージョンの自動削除 679
カンバセーション 679
会話テーブルビュー 680
コンシューマまたはプロバイダーの選択 680
カンバセーションフィルタ 681
観測結果の確認 683
ホバーによるカンバセーション観測 683
フィルタリング 684
会話チャートビュー 684
カンバセーションにおける上位のコンシューマとプロバイダー 685
自動ポリシー検出用の自動ロードバランサ設定（F5 のみ） 687

用語 688
展開（Deployment） 688
クラスタ 689
ポリシー 690
警告 692
ポリシーパブリッシャ 692
前提条件 692
Kafka クライアント証明書の取得 693
Protobuf 定義ファイル 694
Secure Workload ネットワークポリシーのデータモデル 694
Secure Workload ネットワーク ポリシー クライアントのリファレンス実装 695

第 8 章 フォレンジック 697
互換性 698
フォレンジックシグナル 699
特権昇格 700
ユーザーログオン 700
ユーザーログオン失敗 700
シェルコード 700
File Access 701
ユーザーアカウント 701
未確認コマンド 701
未確認ライブラリ 702
raw ソケットの作成 702
バイナリ変更 702
ライブラリ変更 703
サイドチャネル 703
ユーザーログオンの追跡 703
移行手順 703
フォレンジック設定 704
フォレンジックルール 704

フォレンジックルールの追加 704
基本的なフォレンジックルールの構成 706
デフォルトの Secure Workload ルール 706
デフォルトの MITRE ATT&CK ルール 708
フォレンジックプロファイル 715
プロファイルの追加 715
プロファイルの編集 716
プロファイルの複製 717
デフォルトプロファイル：Secure Workload プロファイル 717
デフォルトのプロファイル：MITRE ATT&CK プロファイル 718
変更ログ：フォレンジック 719
法医学の可視化 720
フォレンジックページへのアクセス 720
フォレンジックイベントの閲覧 722
フォレンジックイベントの検査 722
フォレンジックイベントに表示されるフィールド 723
共通のフィールド 724
Process Info 724
特権昇格 724
ユーザーログオン 725
ユーザーログオン失敗 726
シェルコード 727
File Access 727
ユーザーアカウント 727
未確認コマンド 728
未確認ライブラリ 728
raw ソケットの作成 728
ライブラリ変更 729
サイドチャネル 729
ユーザーログオンの追跡 729
移行手順 729

ネットワーク異常 729
フォレンジック分析：検索可能なフィールド 730
その他のフィールド 730
フォレンジック分析の検索用語 730
共通のフィールド 730
バイナリ変更 731
File Access 731
移行手順 731
ユーザーログオンの追跡 732
Ldap 732
ライブラリ変更 732
特権昇格 732
Process Info 733
Raw Socket 733
シェルコード 733
サイドチャネル 734
未確認コマンド 734
未確認ライブラリ 735
ユーザーアカウント 735
ユーザーログオン 735
ユーザーログオン失敗 736
フォレンジックアラート 737
フォレンジックアラートへのアクセス 738
アラート詳細の確認 738
外部との統合 739
フォレンジックスコア 740
フォレンジックスコアが表示される場所 740
フォレンジックスコアの計算方法 741
フォレンジックスコアの改善方法 742
警告 742
PCR ベースのネットワーク異常検出 742

ネットワーク異常イベントのフォレンジックルール 743
ルールの属性 743
ルール アクション 746
ネットワーク異常イベントが表示される場所 746
ルールの重大度とネットワーク異常スコア 748
PCR データとネットワーク異常イベントの保持 749
ネットワーク異常の遅延 749
警告 749
プロセスハッシュの異常検出 749
プロセスハッシュ機能を有効にする方法 751
プロセスハッシュスコアが表示される場所 751
プロセスハッシュスコアの計算方法 752
プロセスハッシュスコアの改善方法 753
脅威情報の詳細 753
警告 754

第 9 章 フロー 757
コーパスセレクタ 758
列とフィルタ 759
フィルタ処理された時系列 765
上位 N 件チャート 767
観測リスト 768
フローの詳細 769
観測結果の確認 770
クライアントサーバーの分類 772
センサータイプの推奨事項 774
フローのプロデューサー（別名サーバー）とコンシューマ（別名クライアント）の識別 775
カンバセーションモード 776

第 10 章 アラート 779

アラートの設定 781
アラートの作成 782
アラート設定モーダル 784
サマリーアラート 786
自動要約とスヌーズの違いに関する注記 787
Cisco Secure Workload Alerts Notifier (TAN) 788
通知機能の設定 788
アラートのパブリッシャの選択 789
外部 syslog トンネリングの TAN への移行 792
接続チャート 793
アラートトリガールールの表示 794
テストアラートの生成 796
現在のアラート 798
アラートのスヌーズ 800
アラート詳細 803
一般的なアラート構造 803
通知機能別の全般アラート形式 808
Kafka（DataTap） 808
E メール 808
PagerDuty 809
Syslog 810
Slack 810
Kinesis 811

第 11 章 メンテナンス 813
サービスのステータス 813
アドミラルアラート 814
アドミラルアラートのライフサイクル 815
個別のアドミラルアラート 816
サマリーのアドミラルアラート 816
アラート詳細 817
ユーザのアクション 821

アドミラル通知 822
クラスタのステータス 824
ファームウェア アップグレードの詳細 828
データのバックアップと復元 830
データ バックアップ 831
データバックアップの前提条件 832
オブジェクトストアの要件 834
データのバックアップの設定 835
ストレージプランナーの使用 836
キャパシティプランナーの使用 837
データバックアップの設定 838
バックアップステータス 841
バックアップスケジュールの非アクティブ化 843
オブジェクトストアの保持 843
チェックポイントの保持 844
データの復元 844
スタンバイモードでのクラスタの展開 844
クラスタデータのプリフェッチ 846
クラスタの復元 847
目標復旧時間と目標復旧時点 848
データのバックアップと復元を行うアップグレード 849
トラブルシューティング：データのバックアップと復元 849
VM の情報 852
クラスタのアップグレード 852
アップグレードの開始 852
RPM アップロード 853
サイト情報 855
アップグレードの事前チェック 856
クラスタのアップグレード 857
ログ 858
アップグレードの事前チェックをいつでも実行 860

Data Backup and Restore（DBR） 861
スナップショット 862
スナップショット作成ユーザーインターフェイスへのアクセス 862
スナップショットの作成 863
CIMC テクニカルサポートバンドルの作成 864
スナップショットの使用 864
デバッグとメンテナンスにスナップショットサービスを使用 868
ランブック 870
Explore / スナップショットのエンドポイントの概要 871
GET コマンド 871
POST コマンド 872
サーバーのメンテナンス 887
ベアメタル除外 895
ディスク メンテナンス 895
クラスタのメンテナンス：クラスタのシャットダウンと再起動 909
クラスタのシャットダウン 909
シャットダウンの開始 909
シャットダウンの進捗状況 910
クラスタの再起動 911
再起動の開始 911
シャットダウンと再起動の履歴 912
[データタップ管理者（Data Tap Admin）]：データのタップ 912

第 12 章 モニタリング 919
エージェントのモニタリング 919
エージェントのモニタリング 919
エージェントのステータスと統計 921
適用ステータス 923
クラウドコネクタの適用ステータス 925
ポリシー更新の一時停止 925
ライセンス 925

ライセンス登録 927
ライセンス使用状況の確認 928
Cisco Smart Licensing の詳細 928

第 13 章 脅威インテリジェンス 931
自動更新 932
手動アップロード 933
更新されたデータセットのダウンロード 933
Cisco Secure Workload へのアップロード 933

第 14 章 レポートダッシュボード 935
レポートダッシュボード 935
概要 935
動作 940
コンプライアンス 943

第 15 章 セキュリティ ダッシュボード 947
セキュリティダッシュボードへの移動 948
セキュリティスコア 948
セキュリティスコアのカテゴリ 948
ハイレベルビュー 949
範囲レベルスコアの詳細 949
総合スコア 949
日次時系列 950
スコアの内訳 951
スコアの詳細 951
脆弱性セキュリティスコア 953
プロセスハッシュスコア 955
攻撃対象領域スコア 957
フォレンジックスコア 961
ネットワーク異常スコア 963

セグメンテーション コンプライアンス スコア 965

第 16 章 脆弱性ダッシュボード 969

脆弱性ダッシュボードへの移動 970

[CVE] タブ 970

[パッケージ（Packages）] タブ 971

[ワークロード（Workloads）] タブ 972

第 17 章 設定 975

ログの変更 975

収集ルール 977

ルール 977

優先順位 977

コレクタ 978

会社 978

アウトバウンド HTTP 接続 979

ログインページのメッセージ 980

セッション設定 980

外部認証の設定 980

Lightweight Directory Access Protocol の設定 983

LDAP 問題のデバッグ 985

LDAP 認証の設定（AD 認証） 986

LDAP 認証の問題のデバッグ 990

シングルサインオン（SSO）の設定 991

ID プロバイダー（IdP）に提供する情報 993

SSO の問題のデバッグ 994

「Use Local Authentication」オプション 994

SSL 証明書およびキー 995

クラスタの設定 997

外部 IPv6 クラスタの接続 998

NTP 認証 999

使用状況分析 1001
[連携（Federation）] 1001
設定 1001
認証設定 1003
管理タスク 1004
スコープ 1004
ワークスペース 1005
ソフトウェアエージェント 1008
その他のタスク 1009
既存の展開 1011
保持できるデータ 1011
保持できないデータ 1013
操作の切断モード 1013
アラート 1013
アラート詳細 1014
API 1016
アプライアンス 1016
アプライアンスオブジェクト 1016
アプライアンスのリスト 1017
スコープ 1017
アイドルセッション 1019
設定 1019
ランディングページ設定の変更 1020
パスワードの変更 1020
パスワードの回復 1020
二要素認証の有効化 1021
二要素認証の無効化 1023
ロール 1023
能力と機能 1025
ロール別のメニューアクセス 1026
新しいロールの作成 1041

ロールの編集 1043
変更ログ：役割 1044
スコープ 1044
テナント 1045
テナントの追加 1045
テナントの編集 1046
ユーザ（Users） 1046
新しいユーザーアカウントの追加 1047
ユーザー アカウントの編集 1049
ユーザーアカウントの非アクティブ化 1051
ユーザーアカウントの再アクティブ化 1051
SSH 公開キーのインポート 1052
Secure Workload セットアップでのサイト設定 1052
ログの変更：ユーザー 1053

第 18 章 OpenAPI 1055
OpenAPI 認証 1056
API キーと秘密の生成 1057
ワークスペースとセキュリティポリシー 1058
ワークスペース 1058
ワークスペースオブジェクト 1058
アプリケーションの一覧表示 1059
単一のワークスペースの取得 1060
ワークスペースの作成 1060
新しいバージョンのインポート 1064
一連のポリシーの検証 1064
ワークスペースの削除 1064
ワークスペースの更新 1065
ワークスペースの詳細を取得 1065
ワークスペースバージョンの一覧表示 1066
ワークスペースバージョンの削除 1067

ワークスペースバージョンの比較 1067
最新のポリシーの分析 1068
単一のワークスペースでポリシー分析を無効にする 1069
単一のワークスペースの適用 1070
単一ワークスペースの適用の無効化 1070
自動ポリシー検出の開始 1071
ポリシー検出実行のステータスの取得 1075
ポリシー 1075
ポリシー オブジェクト 1076
ポリシーの取得 1077
特定のポリシーの取得 1080
ポリシー識別子を使用した特定のポリシーの検索 1080
ポリシーの作成 1082
ポリシーの更新 1083
ポリシーへのサービスポートの追加 1084
ポリシーのサービスポートの更新 1085
ポリシーのサービスポートの削除 1085
ポリシーの削除 1086
識別子を持つポリシーの削除 1086
ポリシーの簡易分析 1087
ポリシー統計情報 1090
未使用のポリシー 1094
ポリシーテンプレート 1096
ポリシーテンプレートの取得 1096
特定のポリシーテンプレートの取得 1097
ポリシーテンプレートの作成 1097
ポリシーテンプレートの更新 1098
ポリシーテンプレートの削除 1099
ポリシーテンプレートのダウンロード 1099
クラスタ 1100
クラスタオブジェクト 1100

クラスタの取得 1101
特定のクラスタの取得 1101
クラスタの作成 1101
クラスタの更新 1103
クラスタの削除 1104
カンバセーション 1105
ポリシー検出実行での会話の検索 1105
ポリシー検出実行の上位 N 件の会話 1107
サポートされているディメンション 1109
サポートされているメトリック 1110
除外フィルタ 1110
除外フィルタオブジェクト 1110
除外フィルタの取得 1111
特定の除外フィルタを取得する 1112
除外フィルタの作成 1112
除外フィルタを更新する 1113
除外フィルタの削除 1114
デフォルトの除外フィルタ 1114
デフォルト除外フィルタのオブジェクト 1115
デフォルトの除外フィルタの取得 1115
特定のデフォルト除外フィルタを取得 1116
デフォルトの除外フィルタの作成 1116
デフォルトの除外フィルタの更新 1117
デフォルトの除外フィルタの削除 1118
ライブ分析 1118
ライブ分析でのフローサイズ 1119
ライブ分析で使用可能なフローメトリック 1119
ライブ分析を介して利用可能なフローのダウンロード 1119
スコープ 1121
範囲オブジェクト 1121
範囲の取得 1122

範囲の作成 1123
特定の範囲を取得する 1124
範囲の更新 1124
特定範囲の削除 1125
ポリシーの優先順位で範囲を取得する 1125
ポリシー順序の更新 1125
範囲クエリの変更のコミット 1126
グループ提案リクエストの送信 1126
グループ提案ステータスの取得 1127
アラート 1127
アラートオブジェクト 1128
アラートの取得 1128
アラートの作成 1129
特定のアラートの取得 1130
アラートの更新 1130
特定のアラートの削除 1130
ロール 1131
ロール オブジェクト 1131
ロールの取得 1131
ロールの作成 1132
特定のロールを取得 1132
ロールの更新 1133
範囲へのロール アクセスを付与 1134
特定のロールの削除 1135
ユーザ（Users） 1135
ユーザー オブジェクト 1135
ユーザーの取得 1136
新しいユーザ アカウントの作成 1137
特定のユーザーの取得 1137
ユーザーの更新 1138
非アクティブ化されたユーザーの有効化/再アクティブ化 1138

ユーザー アカウントにロールを追加 1139
ユーザー アカウントからロールを削除 1139
指定されたユーザーの削除 1140
インベントリ フィルタ 1140
インベントリ フィルタ オブジェクト 1141
インベントリ フィルタの取得 1141
インベントリ フィルタの作成 1142
インベントリフィルタクエリの検証 1143
特定のインベントリ フィルタの取得 1143
特定のインベントリフィルタの更新 1143
特定範囲の削除 1144
フロー検索 1144
フローの寸法のクエリ 1144
フローメトリックのクエリ 1145
フローのクエリ 1145
フィルタ 1147
プリミティブ フィルタ タイプ 1148
論理フィルタ タイプ 1149
フローの TopN クエリ 1149
フローカウント 1151
インベントリ 1152
インベントリの寸法のクエリ 1153
インベントリ検索 1153
インベントリ統計情報 1155
インベントリ カウント 1155
インベントリの脆弱性 1156
ワークロード 1157
ワークロードの詳細 1157
ワークロードの統計情報 1160
インストールされたソフトウェアパッケージ 1161
ワークロードの脆弱性 1161

ワークロードの長時間実行プロセス 1163
ワークロード プロセス スナップショットのサマリー 1164
ワークロード プロセス スナップショット 1165
JSON オブジェクトの定義： 1166
デフォルトのポリシー生成設定 1167
ポリシー生成設定オブジェクト 1168
デフォルトのポリシー生成設定の取得 1169
デフォルトのポリシー生成設定の設定 1170
フォレンジックインテント 1170
フォレンジック インテント オブジェクト 1170
フォレンジックインテントのリスト 1171
単一のフォレンジックインテントの取得 1171
フォレンジックインテントの作成 1171
フォレンジックインテントの更新 1172
フォレンジックインテントの削除 1172
フォレンジックインテント順序 1172
フォレンジックインテント順序オブジェクト 1173
現在のフォレンジックインテント順序の取得 1173
フォレンジックインテント順序の作成 1173
フォレンジックプロファイル 1174
フォレンジック プロファイル オブジェクト 1174
フォレンジックプロファイルのリスト 1175
単一のフォレンジックプロファイルの取得 1175
フォレンジックプロファイルの作成 1175
フォレンジックプロファイルの更新 1175
フォレンジックプロファイルの削除 1176
フォレンジックルール 1176
フォレンジック ルール オブジェクト 1177
フォレンジックルールのリスト 1177
単一のフォレンジックルールの取得 1177
フォレンジックルールの作成 1178

フォレンジックルールの更新 1178
フォレンジックルールの削除 1179
プラットフォーム設定 1179
証明書の取得 1179
使用状況分析設定の取得 1180
ログインメッセージの取得 1180
アウトバウンド HTTP 設定の取得 1180
アウトバウンド HTTP 設定の更新 1181
アウトバウンド HTTP 設定のテスト 1181
アウトバウンド HTTP プロキシ設定の取得 1181
アウトバウンド HTTP プロキシ設定の更新 1182
施行 1183
エージェント ネットワーク ポリシーの設定 1183
具体的なポリシーの統計情報 1184
JSON オブジェクトの定義： 1184
クライアントサーバー構成 1192
ホスト構成 1192
ポート設定 1194
ソフトウェアエージェント 1196
エージェント API 1196
インテントを使用したソフトウェア エージェントの設定 1197
インターフェイス構成インテント 1202
NAT の背後にあるエージェントの VRF 設定 1203
Cisco Secure Workload ソフトウェアのダウンロード 1204
サポートされているプラットフォームを取得するための API 1204
サポートされているソフトウェアバージョンを取得するための API 1205
インストーラ ID を作成する API 1206
Secure Workload ソフトウェアをダウンロードするための API 1206
Cisco Secure Workload エージェントのアップグレード 1207
エージェントを特定のバージョンにアップグレードするための API 1207
ユーザーがアップロードしたファイルハッシュ 1208

ユーザーファイルハッシュのアップロード 1208
ユーザーファイルハッシュの削除 1209
ユーザーファイルハッシュのダウンロード 1210
ユーザー定義ラベル 1210
範囲依存 API 1211
範囲に依存しない API 1217
範囲に依存しないラベル 1218
Virtual Routing and Forwarding（VRF） 1221
VRF オブジェクト 1221
VRF の取得 1222
VRF を作成します 1222
特定の VRF の取得 1223
VRF の更新 1223
特定の VRF の削除 1224
オーケストレーション 1225
オーケストレーション オブジェクト 1225
Ingress コントローラ 1228
ポッドセレクタ 1228
コントローラ設定 1228
Infoblox 設定 1229
オーケストレーションの取得 1230
オーケストレーションの作成 1230
特定のオーケストレーションを取得する 1232
オーケストレーションの更新 1232
特定のオーケストレーションの削除 1232
オーケストレータのゴールデンルール 1232
オーケストレータのゴールデン ルール オブジェクト 1232
オーケストレータのゴールデンルールを取得する 1233
ゴールデンルールの作成/更新 1233
FMC オーケストレータドメイン 1233
オーケストレータ FMC ドメインオブジェクト 1234

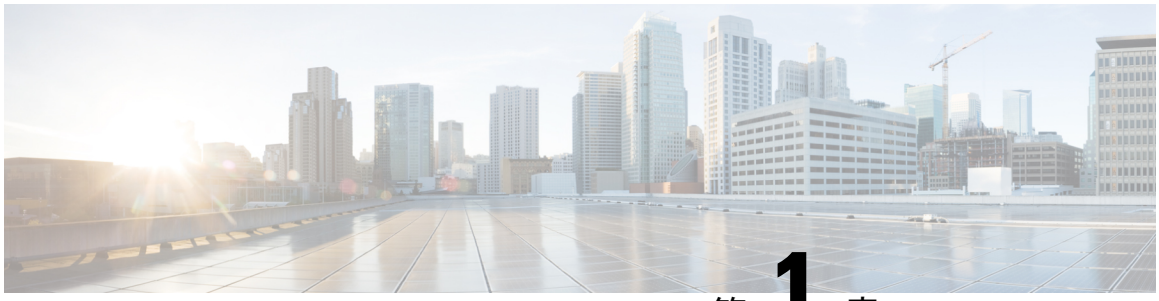
FMC ドメインの取得 1234
FMC 外部オーケストレータの FMC ドメイン設定の更新 1235
RBAC（役割ベースのアクセス制御）に関する考慮事項 1236
高可用性とフェールオーバーに関する考慮事項 1236
Kubernetes RBAC リソースに関する考慮事項 1236
Service Health 1238
サービス正常性の取得 1238
Secure Connector 1238
ステータスの取得 1238
トークンを取得 1239
証明書のローテーション 1239
外部オーケストレータのポリシー適用ステータス 1240
すべての外部オーケストレータのポリシー適用ステータスを取得する 1240
外部オーケストレータのポリシー適用ステータスの取得 1240
管理対象データタップとデータシンクの証明書のダウンロード 1241
特定の VRF ID に対するマネージド DataTap のリストの取得 1241
指定された MDT ID に対するマネージド DataTap 証明書のダウンロード 1241
特定の VRF ID に対する DataSink のリストの取得 1242
指定された DataSink ID の DataSink 証明書をダウンロードします。 1242
変更ログ 1243
変更ログオブジェクト 1243
検索 1243
ルーティング不可能なエンドポイント 1245
ルーティング不可能なエンドポイントオブジェクト 1245
ルーティング不可能なエンドポイントの取得 1246
ルーティング不可能なエンドポイントの作成 1246
名前を使用したルーティング不可能なエンドポイントの取得 1247
ID を使用したルーティング不可能な特定のエンドポイントの取得 1247
ルーティング不可能な特定のエンドポイント名の更新 1247
名前を使用したルーティング不可能なエンドポイントの削除 1248
名前を使用したルーティング不可能なエンドポイントの削除 1248

外部アプライアンスおよびコネクタの設定およびコマンドのスキーマ 1248
設定グループ API 1248
設定のスキーマを取得する API 1248
トラブルシューティング コマンドのスキーマを取得する API 1250
外部アプライアンス 1251
外部アプライアンス API 1251
コネクタ 1267
コネクタ API 1267

第 19 章 制限 1293
フローとエンドポイント 1293
テナント、子範囲、インベントリフィルタ、およびロール 1293
Cloud Connector 1294
コネクタ 1295
コネクタ用の Cisco Secure Workload 仮想アプライアンス 1296
ラベルの制限 1297
ポリシーに関連する制限 1298
その他の機能 1300
データイン/データアウト 1305

第 20 章 Cisco Secure Workload Virtual（SECURE-WORKLOAD-V） 1307
Cisco Secure Workload Virtual 1307

第 21 章 エンド ユーザ ライセンス契約書 1309
エンド ユーザ ライセンス契約書 1309



第 1 章

はじめに

- Cisco Secure Workload の概要 （1 ページ）
- クイックスタートウィザード （2 ページ）
- セグメンテーションとマイクロセグメンテーションを使用する前に （3 ページ）

Cisco Secure Workload の概要

現代のネットワークには、ベアメタル、仮想化、クラウドベース、およびコンテナベースのワークロードを使用し、ハイブリッドなマルチクラウド環境で実行されるアプリケーションが含まれています。重要な課題は、ネットワークの俊敏性を損なうことなく、アプリケーションとデータをより安全に保護する方法を実現することです。Cisco Secure Workload（旧称 Cisco Tetration）は、アプリケーションにふさわしいセキュリティを提供し、アプリケーションの動作に基づいてセキュリティ態勢を調整することで、包括的なワークロード保護を実現し、このセキュリティの課題に対処できるように設計されています。Secure Workload は、高度な機械学習および行動分析技術を使用してこれを達成します。プラットフォームには、次のセキュリティユースケースをサポートする、すぐに使用できるソリューションが用意されています。

- ゼロトラストモデルの実装を許可するマイクロセグメンテーションポリシー：ビジネス目的で必要なトラフィックのみを許可するポリシーを適用します。
- ワークロードの行動的ベースライン、分析、および異常の特定。
- 一般的な脆弱性とサーバーにインストールされたソフトウェアパッケージに関連したエクスポージャーの検出。
- 脆弱性の検出時にサーバーをプロアクティブに隔離し、通信をブロックするポリシーを適用可能。

ワークロードについて

ワークロードは IP アドレスです。

この製品では、「IPアドレス」と呼ばれる Secure Workload エージェントがインストールされていないホストと区別するために、「ワークロード」は特にエージェントがインストールされているホストを指す場合があります。

クイックスタートウィザード

オプションのウィザードを使用して、セグメンテーションを開始できます。このウィザードは、範囲ツリーの最初のブランチの作成方法を示し、この階層構造とそれを形成するラベルの強力なメリットを紹介します。または、エージェントをインストールしてデータ収集を開始して、事業運営に必要なポリシーを Secure Workload で自動的に作成することもできます。

次のユーザーロールがこのウィザードにアクセスできます。

- サイト管理者
- カスタマーサポート
- ルート範囲の所有者

このウィザードにアクセスするには、次の手順を実行します。

- [整理（Organize）] > [範囲とインベントリ（Scopes and Inventory）] で範囲を定義することはできません

  （範囲が既に作成されている場合は、既存の範囲をすべて削除しない限り、このウィザードに再度アクセスすることはできません）。
- Cisco Secure Workload にサインインすると、このウィザードが表示されます。
- または、任意のページの上部で、青いバナー内のリンクをクリックします。
- または、ウィンドウの左側にあるメインメニューから [概要（Overview）] を選択します。

詳細情報：

- *Cisco Secure Workload 3.8 Quick Start Guide*（https://cisco.com/go/secure-workload-quick-start-guide-38） [英語]
- ワークロードラベル
- 範囲とインベントリ
- ソフトウェアエージェントの展開
- ポリシー

セグメンテーションとマイクロセグメンテーションを使用する前に

以下の手順は非常に高度なものです。Secure Workloadを使用してセグメンテーションとマイクロセグメンテーションのポリシーを設定するための出発点として使用してください。

マイクロセグメンテーション導入の一般的なプロセス

セグメンテーションとマイクロセグメンテーションの目的は、組織がビジネスを遂行するために必要なトラフィックのみを許可し、他のすべてのトラフィックをブロックすることです。

手順

ステップ1 要件を満たす。

ワークロードが実行されているプラットフォームとバージョン、およびポリシーに通知する重要な情報を提供するシステムをSecure Workloadがサポートしていることを確認します。

https://www.cisco.com/go/secure-workload/requirements/agentsを参照してください。

ステップ2 ワークロードにエージェントをインストールします。

エージェントは、ワークロードをグループ化し適切なポリシーを決定するためにユーザーとSecure Workloadによって使用されるフローデータやその他の情報を収集します。その後準備ができた時点で、これらのエージェントは、ユーザーが承認したポリシーを適用します。サポートされているプラットフォームと要件のリストへのリンクを含む詳細については、「ソフトウェアエージェントの展開」を参照してください。

ステップ3 ワークロードを説明するラベルを収集するかアップロードします。

ラベルを使用すると、各ワークロードの目的やその他の重要な情報を容易に把握できるようになります。

この情報は、ワークロードをグループ化し、適切なポリシーを適用し、Secure Workloadによって提案されたポリシーを理解するために必要です。ラベルは、長期的なポリシー管理を簡素化するセルフメンテナンスグループの基盤です。詳細については、「ワークロードラベル」とカスタムラベルのインポート （414 ページ）を参照してください。

ステップ4 ワークロードのラベルに基づいて範囲ツリーを作成します。

ラベルが作成に役立つワークロードの論理グループは「範囲」と呼ばれ、適切に選択されたラベルのセットは、範囲ツリーと呼ばれるネットワークの階層型マップを作成するのに役立ちます。ネットワーク上のワークロードの階層ビューは、ポリシーを効率的に作成およびメンテナンスするための鍵です。このビューを使用すると、ポリシーを1度作成するだけで、そのポリシーをツリーの該当するブランチのすべてのワークロードに自動的に適用できます。また、特

定のアプリケーション（またはネットワークの一部）に関する責任を、それらのワークロードの正しいポリシーを決定するために必要な専門知識を持つ人員に委任することもできます。

ラベルに基づくクエリに基づいて、ワークロードを範囲にグループ化します。たとえば、「Application＝Email-app」および「Environment＝Production」というラベルを持つすべてのワークロードを含む「Email-app」という範囲を作成できます。「Environment＝Production」というクエリを使用して、この範囲の親範囲を作成できます。Production（実稼働）範囲には、実稼働 Emailapp と、「Environment＝Production」とラベル付けされた他のすべてのワークロードが含まれます。

詳細については、「範囲とインベントリ」を参照してください。

範囲をまだ作成していない場合は、クイックスタートウィザードを使用して最初の範囲ツリーを作成できます。クイックスタートウィザード （2 ページ）を参照してください。

ステップ 5 ポリシーを作成する範囲ごとにワークスペースを作成します。

ワークスペースは、範囲内のワークロードのポリシーを管理する場所です。詳細については、「ワークスペースを使用したポリシーの管理」およびサブトピックを参照してください。

ステップ 6 ネットワーク全体に広く適用されるポリシーを手動で作成します。

たとえば、すべての内部ワークロードから NTP サーバーへのアクセスを許可し、すべての外部トラフィックを拒否するか、明示的に許可されていない限り、すべての非内部ホストからのアクセスを拒否することができます。より詳細に適用されるポリシーでオーバーライドできない絶対ポリシーと、より具体的なポリシーが存在する場合にオーバーライドできるデフォルトポリシーを作成できます。通常、これらのポリシーは、ツリーの最上部に近い範囲に関連付けられたワークスペースで作成します。

ポリシーの手動作成 （525 ページ）を参照してください。

- （オプション）ニーズに対応するテンプレートを確認します。ポリシーテンプレート （529 ページ）を参照してください。
- （オプション）ポリシーを検出する複雑なプロセスを開始する前に、手動で作成したポリシーを、期待する効果が発揮されることが確認できたらすぐに適用します。

  ポリシーの適用 （654 ページ）を参照してください。

ステップ 7 既存のトラフィックパターンに基づいて、ポリシーを自動的に検出します。

Secure Workload は、ワークロード間のトラフィックを分析し、動作に基づいてワークロードをグループ化し、組織が必要とするトラフィックを許可するための一連のポリシーを提案するため、他のすべてのトラフィックをブロックできます。

一般に、長期間にわたるフローデータが多いほど、より正確なポリシー提案が作成されます。

ポリシーは繰り返し検出できます（これについては、この手順で後に詳しく説明します）。

次の操作を実行できます。

1. （オプション）範囲ツリーのブランチの粗いポリシーを検出します。

使用を始めたばかりの場合は、一時的な一連のポリシーをすばやく配置し、将来の脅威からある程度保護することができます。

2. 単一範囲のポリシーを検出します。

   通常、これは範囲ツリーの最下位またはその近くにある範囲に対して行います。これらの範囲には、通常、単一アプリケーションのワークロードが含まれます。

詳細については、「ポリシーの自動検出」およびサブトピック（1 つの範囲または範囲ツリーのブランチのポリシーの検出 （538 ページ）を含む）を参照してください。

ステップ 8 ポリシーを確認して分析します。

ポリシーを慎重に調べて、期待どおりの効果があり、意図していない効果がないことを確認します。

組織内の対象分野の専門家およびアプリケーションオーナーと協力して、組織のニーズと提案されたポリシーの適切性を判断してください。

a) Secure Workload から提案されたポリシーとクラスタを確認し、理にかなっていることを確認します。

   （クラスタは 1 つの範囲内で密接に関連しているワークロードのグループであり、クラスタでは、範囲全体を対象としたポリシーよりもカスタマイズされたポリシーの方が正当とされる可能性があります。ワークロードのグループ化：クラスタとインベントリフィルタ （595 ページ）およびサブトピックを参照してください）。

   自動検出されたポリシーの確認 （633 ページ）を参照してください。

b) ポリシーを分析して、ネットワーク上の実際のトラフィックにどのように影響するかを確認します。

   Secure Workload でポリシー分析やその他のツールを使用して、組織がビジネスを実施するために必要なトラフィックがポリシーに許可されていることを確認します。

   「ライブポリシー分析」とポリシーの視覚的表現 （637 ページ）を参照してください。

   ポリシーの結果を分析する際は、次の点に注意してください。

   - 階層内で各範囲の上位にある範囲のワークスペース内のポリシーが、同じブランチの下位にある範囲のワークロードに影響を与える可能性があります。ポリシーの継承と範囲ツリー （522 ページ）を参照してください。
   - マイクロセグメンテーションは、各ワークロードの周囲に小規模なファイアウォールを配置します。接続を成功させるには、トランザクションのコンシューマとプロバイダーの両方に、トラフィックを許可するポリシーが必要です。両方のワークロードが同じ範囲にない場合は、これらのポリシーを作成するために、追加の手順が必要になる場合があります。コンシューマとプロバイダーが異なる範囲にある場合：ポリシーオプション （615 ページ）を参照してください。

ステップ 9 必要に応じて繰り返しポリシーを検出します。

トラフィックフローデータが多いほどポリシーの提案がより正確になるため（たとえば、毎月実行されるレポートがある場合、3 週間分のデータでもすべての重要なトラフィックをキャプチャできていない可能性があります）、ポリシーの検出を継続し、新たに検出されたポリシー提案を確認し分析できます。検出を実行するたびに、その時点における既存のトラフィックフローに基づいてポリシーが提案されます。

また、設定されたポリシー検出設定や承認されたクラスタなどの変更をキャプチャするために繰り返すこともできます。

「反復的なポリシーの変更 （568 ページ）」およびサブトピックを参照してください。

自動ポリシー検出を再実行する前に、保持するポリシーとクラスタの承認などの、サブトピックで説明されている重要な手順を実行する必要があります。

ポリシーを再検出するたびに、ポリシーを再確認および再分析する必要があります。

ステップ 10 準備ができたら、ポリシーを適用します。

ワークスペース（および関連付けられた範囲）に関連付けられたポリシーが適切であり、重要なサービスを中断せずに不要なトラフィックをブロックすると判断したら、それらのポリシーを適用できます。

ポリシーは繰り返し適用できます。たとえば、最初はツリーの最上位近くの範囲で手動で作成したポリシーのみを適用し、その後、ツリーの下位の範囲で検出済みのポリシーを適用していくことができます。

詳細については、ポリシーの適用 （654 ページ）およびサブトピックを参照してください。

ベアメタルまたは仮想マシンで実行されるワークロードのマイクロセグメンテーションを設定する

手順

ステップ 1 ネットワーク上のワークロードの IP アドレスの収集を開始します。

ワークロードごとに、アプリケーション名、アプリケーションの所有者、環境（本番または非本番）、および適用するポリシーを決定するその他の情報（地理的地域など）も必要になります。

構成管理データベース（CMDB）がない場合は、この情報をスプレッドシートで収集できます。

開始するには、焦点を合わせる単一のアプリケーションを選択します。

ステップ 2 サポートされているベアメタルベースまたは仮想ワークロードにエージェントをインストールします。

「ソフトウェアエージェントの展開」を参照してください。

ステップ 3 これらのワークロードを説明するラベルをアップロードします。

「ワークロードラベル」およびカスタムラベルのインポート（414 ページ）を参照してください。

必要に応じて、初回ウィザードを実行して、ラベルと範囲ツリーの最初のブランチを作成できます。ウィザードの詳細については、「クイックスタートウィザード」を参照してください。

ステップ 4 必要に応じて、ラベルに基づいて範囲ツリーを作成または更新します。

「範囲とインベントリ」を参照してください。

ステップ 5 ポリシーを適用する範囲ごとにワークスペースを作成します。

「ワークスペースを使用したポリシーの管理」およびサブトピックを参照してください。

ステップ 6 ネットワーク全体に広く適用される手動ポリシーを作成します。

ポリシーの手動作成 （525 ページ）を参照してください。

ステップ 7 プラットフォーム固有のポリシー （578 ページ）の該当する追加情報を参照してください。

ステップ 8 下位レベルの範囲に関連付けられたワークスペースのポリシーを自動的に検出します。

「ポリシーの自動検出」およびサブトピックを参照してください。

ステップ 9 提案されたポリシーを確認して分析します。

「ポリシーの確認と分析 （633 ページ）」およびサブトピックを参照してください。

ステップ 10 必要に応じて繰り返しポリシーを検出します。

「反復的なポリシーの変更 （568 ページ）」およびサブトピックを参照してください。

ステップ 11 準備ができたら、ポリシーを適用します。

各ワークスペースのポリシーの動作に問題がなければ、ポリシーを適用できます。

ワークスペースとエージェント設定の両方でポリシーの適用を有効にする必要があります。

詳細については、ポリシーの適用 （654 ページ）およびサブトピックを参照してください。

クラウドベースのワークロードに対するマイクロセグメンテーションの設定

手順

ステップ 1 （オプション）クラウドベースのワークロードにエージェントをインストールします。

Cloud Connector が提供する VPC/VNet レベルの粒度よりも細かいレベルでポリシーの検出と適用が必要な場合は、サポートされているプラットフォームにエージェントをインストールします。

クラウドサービスが実行されているオペレーティングシステムに基づいて、エージェントをインストールします。「ソフトウェアエージェントの展開」を参照してください。

ステップ 2 Cloud Connector を設定して、ラベルとフローデータを収集します。

その場合は、次のトピックを参照してください。

- AWS コネクタ。
- Azure コネクタ。

ステップ 3 コネクタによって作成された範囲のワークスペースを作成します。

「ワークスペースを使用したポリシーの管理」を参照してください。

ステップ 4 ポリシーを自動的に検出します。

VPC/VNet で定義された範囲ごとに（該当する場合は、より細かい範囲ごとに）ポリシーを検出します。

「ポリシーの自動検出」およびサブトピックを参照してください。

ステップ 5 提案されたポリシーを確認して分析します。

「ポリシーの確認と分析 （633 ページ）」およびサブトピックを参照してください。

ステップ 6 必要に応じて繰り返しポリシーを検出します。

「反復的なポリシーの変更 （568 ページ）」およびサブトピックを参照してください。

ステップ 7 準備ができたら、各範囲のポリシーを承認して適用します。

該当するワークスペースと各 VPC または VNet のコネクタ、および/または個々のワークロードにインストールされているエージェントの適用を有効にする必要があります。

- 詳細については、ポリシーの適用 （654ページ）およびサブトピックを参照してください。
- AWS ベースのワークロードについては、「AWS インベントリにセグメンテーションポリシーを適用するときのベストプラクティス」を参照してください。
- Azure ベースのワークロードについては、「Azure Inventory にセグメンテーションポリシーを適用するときのベストプラクティス」を参照してください。

Kubernetes ベースのワークロードに対するマイクロセグメンテーションの設定

手順

ステップ 1 Kubernetes ベースのワークロードにエージェントをインストールします。

要件および前提条件を確認してください。

「優れた可視性と適用のための Kubernetes または OpenShift エージェントのインストール」を参照してください。

エージェントは、該当する Kubernetes サービスによって管理される将来のすべてのワークロードに自動的にインストールされます。

ステップ 2 Kubernetes ベースのワークロードのラベルを収集します。

Kubernetes の展開に応じて、以下を参照してください。

- シンプルな Kubernetes およびオープンソースワークロードの場合は、以下を参照します。

  外部オーケストレータ （163 ページ）および Kubernetes/OpenShift （184 ページ）

- Amazon Web Services（AWS）で実行される Elastic Kubernetes Service（EKS）の場合は、以下を参照します。

  AWS コネクタ （290 ページ）および AWS（EKS）で実行されるマネージド Kubernetes サービス （305 ページ）

- Azure Kubernetes Services（AKS）の場合は、以下を参照します。

  Azure コネクタ （308 ページ）および Azure（AKS）で実行されるマネージド Kubernetes サービス （317 ページ）

- Google Cloud Platform（GCP）で実行される Google Kubernetes Engine（GKE）の場合は、以下を参照します。

  GCP（GKE）で実行されるマネージド Kubernetes サービス （327 ページ）

ステップ 3 ラベルに基づいて範囲ツリーを作成または更新します。

「範囲とインベントリ」を参照してください。

ステップ 4 ポリシーを適用する範囲ごとにワークスペースを作成します。

「ワークスペースを使用したポリシーの管理」を参照してください。

ステップ 5 各低レベル範囲のポリシーを自動的に検出します。

「ポリシーの自動検出」を参照してください。

ステップ 6 （オプション）プラットフォーム固有のポリシー（578ページ）の該当する追加オプションを参照してください。

ステップ 7 提案されたポリシーを確認して分析します。

ポリシーの確認と分析 （633 ページ）を参照してください。

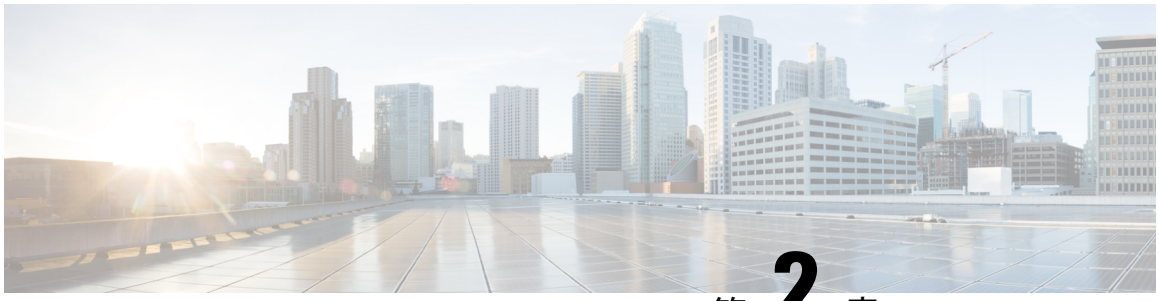
ステップ 8 必要に応じて、ポリシーを繰り返し検出、レビュー、分析します。

反復的なポリシーの変更 （568 ページ）を参照してください。

ステップ 9 準備ができたら、各範囲のポリシーを承認して適用します。

ワークスペースとエージェントでポリシーの適用を有効にする必要があります。

ポリシーの適用 （654 ページ）とサブトピック（コンテナへの適用 （663 ページ）を含む）を参照してください。



第 2 章

Cisco Smart Licensing

シスコのスマート ライセンスは、複数のシスコ製品間でソフトウェア ライセンスを管理する統合ライセンス管理システムです。Cisco Smart Licensing アカウントを持っている場合は、Cisco Smart Licensing トークンを Cisco Secure Workload ライセンスに関連付けることができます。

Cisco Secure Workload ライセンス

- **Tetration ワークロード保護**：ワークロード保護用。ライセンス情報は、[ライセンスの使用状況情報（License Usage Information）] ページの [合計ワークロードライセンス使用状況（Total workload license usage）] テーブルで確認できます。
- **Tetration エンドポイントの可視性**：Cisco Secure Workload でのワークロードの可視性用。消費されたワークロードとライセンスの数は、[ライセンスの使用状況情報（License Usage Information）] ページの [合計エンドポイントライセンス使用状況（Total endpoint license usage）] テーブルで確認できます。
- **Cisco Secure Workload PLR**：Cisco Secure Workload でライセンス予約 PLR モードを使用するには、PLR ライセンスを購入します。

Cisco Secure Workload スマートライセンスの登録

表 *1 : Cisco Secure Workload* スマートライセンスの登録

新規ユーザ	既存ユーザー
1. Cisco Software Central でスマートアカウントを作成し、Cisco Secure Workload の利用資格を購入します。詳細については、「Cisco Licensing」を参照してください。 2. Smart Software Manager に移動し、Cisco Secure Workload クラスタを登録するためのトークンを作成します。 3. Cisco Secure Workload から Cisco Smart Software Manager（CSSM）ポータルに接続して、クラスタを登録し、すべてのライセンスとコンプライアンス情報を同期します。また、特定のライセンスを予約することもできます。	1. 購入した Cisco Secure Workload の利用資格は、Cisco Smart Software Manager でスマートアカウントに保存されます。管理するには、Smart Software Manager に移動します。 2. Cisco Secure Workload リリース 3.8 にアップグレードします。 3. Smart Software Manager に移動し、Cisco Secure Workload クラスタを登録するためのトークンを作成します。 4. 使用している最新バージョンの Cisco Secure Workload から Cisco Smart Software Manager（CSSM）ポータルに接続して、クラスタを登録し、すべてのライセンスとコンプライアンス情報を同期します。また、特定のライセンスを予約することもできます。

スマートアカウントの作成の詳細については、次を参照してください。

- スマートアカウントの作成方法
- スマートアカウントの作成方法 - ビデオ

Cisco Secure Workload で使用可能なライセンスオプション：

- **接続モード**：利用資格とコンプライアンス情報が、Cisco Secure Workload と Smart Software Manager によって定期的に伝達されます。Smart Software Manager で変更を加えた場合は、Cisco Secure Workload 上で認証を更新すると、その変更がすぐに適用されます。また、スケジュールどおりに Cisco Secure Workload が通信するのを待つこともできます。Cisco Secure Workload スマートライセンスの登録：CSSM ポータル （13 ページ）を参照してください。
- **エアギャップモード**
  - Smart Software Manager オンプレミス：Smart Software Manager オンプレミスを使用することで、Smart Software Manager でスマートライセンス認証の同期をスケジュールしたり、手動で行ったりできます。Smart Software Manager オンプレミスを使用して Cisco Secure Workload クラスタを登録するには、Cisco Secure Workload スマートライセンスの登録：CSSM オンプレミス （20 ページ）を参照してください。

- 特定のライセンスの予約：Smart Software Manager にアクセスしたり、Smart Software Manager オンプレミスを使用したりせずに仮想スマートアカウントからライセンスを予約するには、ライセンスの予約の方法を使用します。ライセンス予約 （16ページ）を参照してください。



（注）

- スマートライセンスは、Cisco Secure Workload のオンプレミスバージョンに適用され、SaaS バージョンには適用されません。
- スマートライセンス機能には、バージョン 3.8 以降で実行されている Cisco Secure Workload クラスタが必要です。
- データのバックアップと復元の機能のために、プライマリ（アクティブ）クラスタにはスマートライセンスが必要ですが、スタンバイクラスタにはスタンバイクラスタがアクティブになるまでは必要ありません。
- フェデレーションクラスタの場合、リーダーとフォロワーはスマートライセンスを使用して個別に登録する必要がありますが、リーダーはエンドポイントまたはワークロードのライセンスを消費しません。
- ライセンスが過剰に使用されている場合、またはライセンスの有効期限が切れている場合、Cisco Secure Workload クラスタは非準拠状態になります。クラスタを準拠状態に復元するためのアクションをすぐに実行することを推奨します。リリース 3.8 では、非準拠状態によって影響を受ける機能はありません。

- Cisco Secure Workload スマートライセンスの登録：CSSM ポータル （13 ページ）
- ライセンス予約 （16 ページ）
- Cisco Secure Workload スマートライセンスの登録：CSSM オンプレミス （20 ページ）
- スマートライセンスの同期 （21 ページ）

Cisco Secure Workload スマートライセンスの登録：CSSM ポータル

始める前に

次の点を確認します。

- Cisco Secure Workload クラスタを CSSM ポータルに登録するために、インターネット接続が使用できる。
- スマート ライセンス アカウントに必要なライセンスが含まれている。

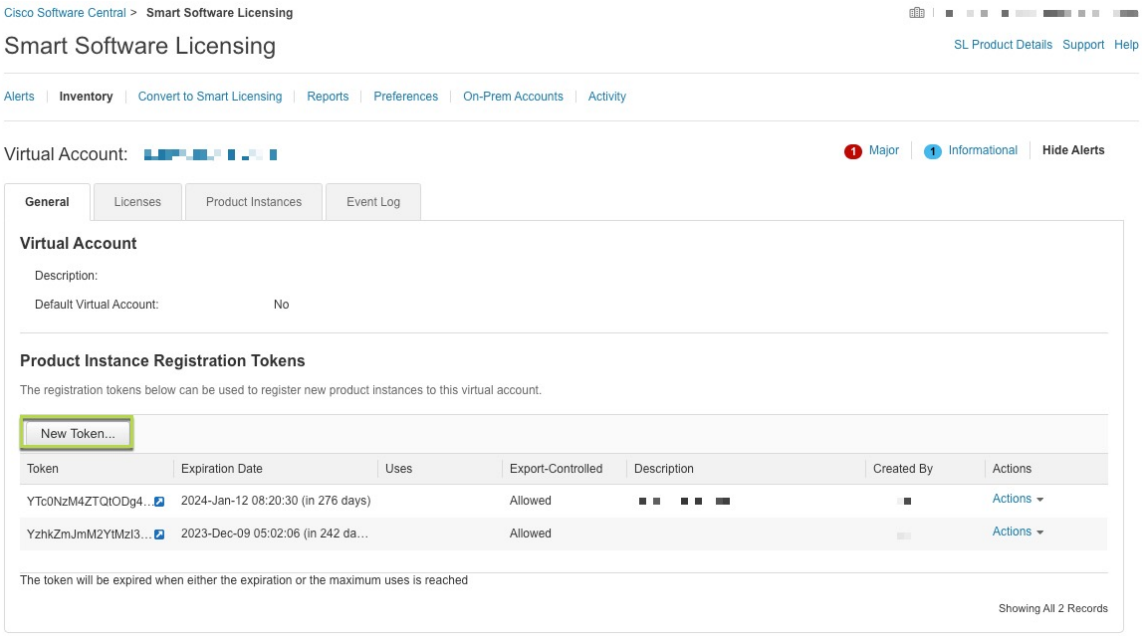
手順

ステップ 1 Cisco Smart Software Manager（CSSM）ポータルに移動し、次のアクションを実行して、Cisco Secure Workload クラスタを登録するバーチャルアカウントのトークンを生成します。

a) CSSM ポータルにログインします。

b) 正しいバーチャルアカウントを選択し、**[スマートソフトウェアライセンシング（Smart Software Licensing）]** > **[一般（General）]** に移動して、[新しいトークン（New Token）] をクリックします。画面の指示に従って、トークンを作成します。

図 *1 : CSSM* ポータルでの新しいトークンの作成

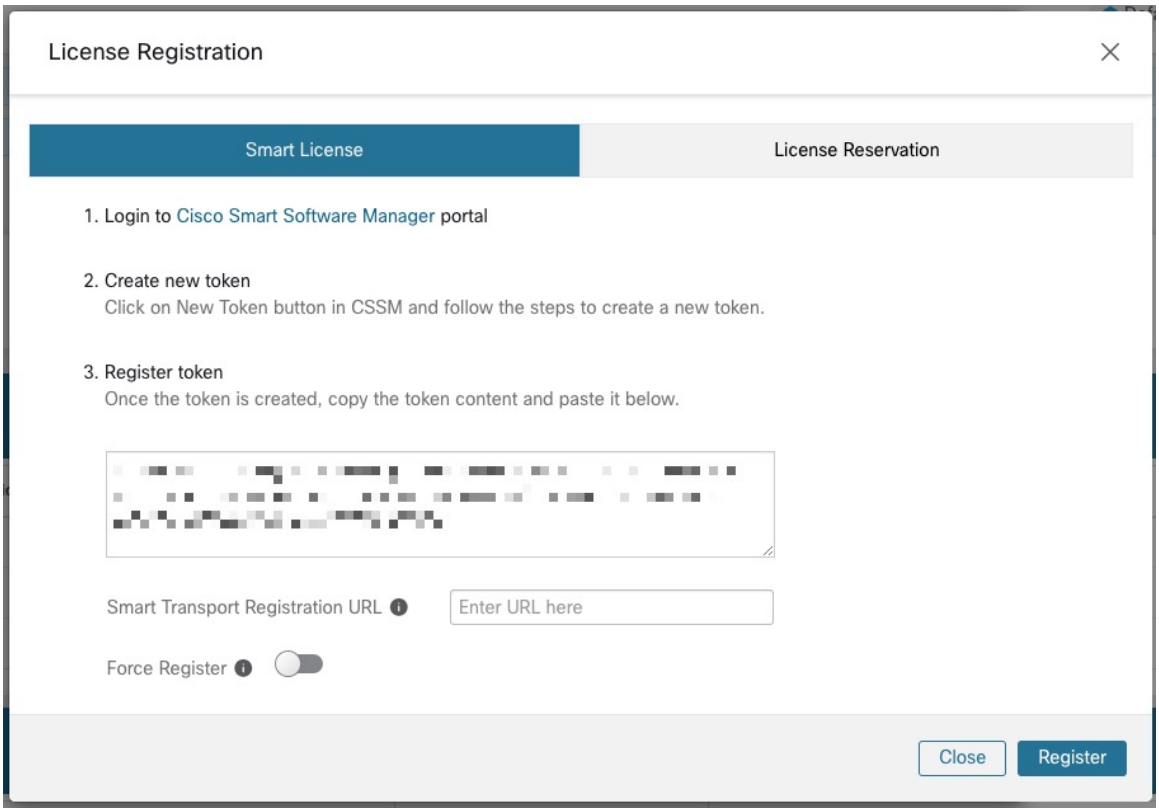


c) 生成されたトークンをコピーします。

ステップ 2 Cisco Secure Workload で、**[管理（Manage）]** > **[サービスの設定（Service Settings）]** > **[ライセンス（Licenses）]** に移動し、[登録（Register）] をクリックします。

ステップ 3 [スマートライセンス（Smart Licenses）] タブで、CSSM ポータルから生成されたトークンを入力します。

図 *2 : Cisco Secure Workload* スマートライセンスの登録ページ



ステップ 4 （オプション）CSSM を使用してクラスタ UUID がすでに登録されている場合でも登録を続行するには、[強制登録（Force Register）] を有効にします。

ステップ 5 [Register] をクリックします。

CSSM ポータルを使用した Cisco Secure Workload の登録が開始されます。登録の更新されたステータスを表示するには、[ライセンスの使用状況情報（License Usage Information）] ページを更新します。

Cisco Secure Workload スマートライセンスの登録解除

手順

ステップ 1 [ライセンスの使用状況情報（License Usage Information）] ページで、[登録解除（Deregister）] をクリックします。

ステップ 2 確認するには、[はい（Yes）] をクリックします。

[ライセンスの使用状況情報（License Usage Information）]ページを更新すると、[ライセンスのステータス（Licensing Status）] フィールドにライセンスのステータスが [未登録（Not Registered）] と表示され、クラスタは評価期間に入ります。

ライセンス予約

ライセンス予約機能を使用して、エアギャップネットワークにスマートライセンスを展開できます。開始する前に、Cisco Secure Workload に必要なライセンスがスマートアカウントで使用可能であることを確認します。アカウントで使用可能なライセンスの数を確認するには、CSSM スマート ソフトウェア ライセンス ポータルに移動し、**[インベントリ（Inventory）]** > **[ライセンス（Licenses）]** をクリックします。

手順

ステップ 1 Cisco Secure Workload UI で、**[管理（Manage）]** > **[サービスの設定（Service Settings）]** > **[ライセンス（Licenses）]** に移動します。
[ライセンスの使用状況情報（License Usage Information）] ページが表示されます。

ステップ 2 [Register] をクリックします。

ステップ 3 [ライセンス予約（License Reservation）] タブを選択し、[はい（Yes）] をクリックして確定します。
要求コードが Cisco Secure Workload によって生成されます。

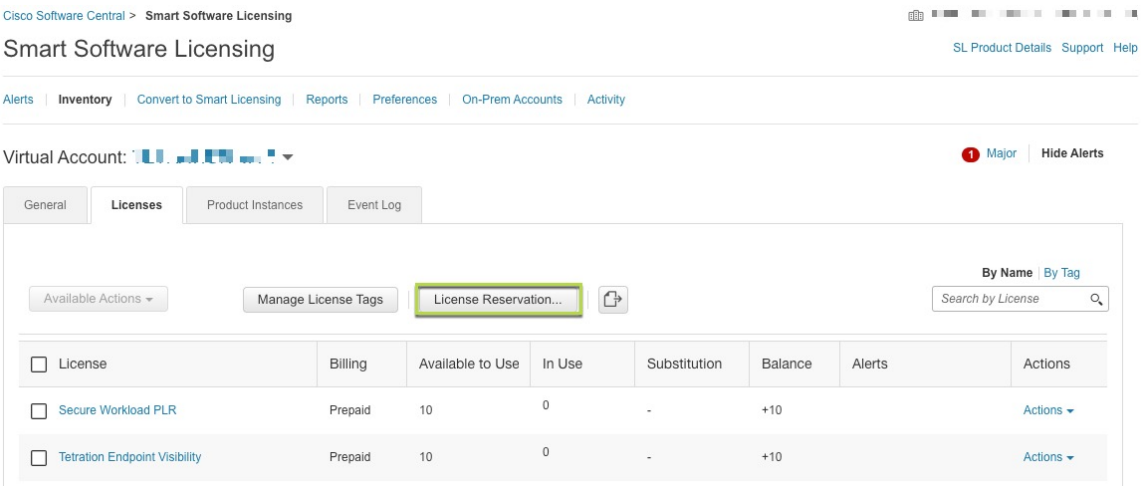
ステップ 4 生成された要求コードをコピーします。

ステップ 5 CSSM スマート ソフトウェア ライセンス ポータルに移動し、次のアクションを実行します。

a) **[インベントリ（Inventory）]** > **[ライセンス（Licenses）]** に移動します。

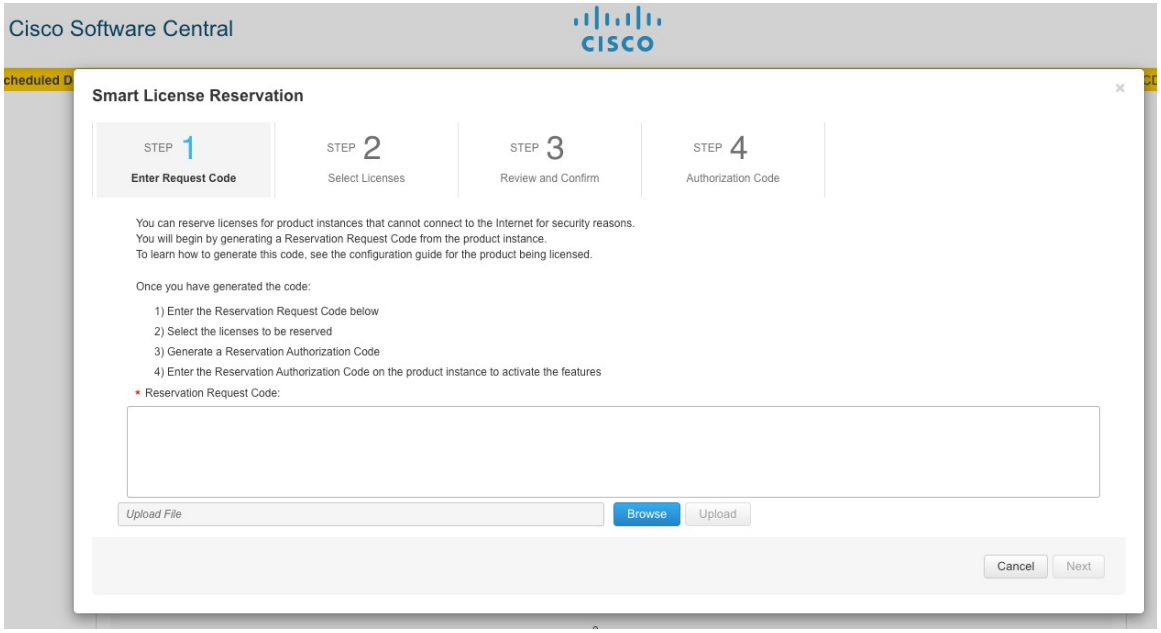
b) [ライセンス予約（License Reservation）] をクリックし、[続行（Proceed）] をクリックして続行します。

図 *3 : [ライセンス（Licenses）]* ページ：*CSSM* ポータル



c) Cisco Secure Workload によって生成された要求コードを入力し、[次へ（Next）] をクリックします。

図 *4*：ライセンス予約：*CSSM* ポータル



d) [予約するライセンス（Licenses to Reserve）] で、次のいずれかを選択し、[次へ（Next）] をクリックします。

- [Cisco Secure Workload PLR（Secure Workload PLR）]：クラスタで設定できるワークロードとエンドポイントの数に制限はありません。
- [特定のライセンスの予約（Reserve a specific license）]：購入したワークロードとエンドポイントのライセンスに基づいて、Tetration ワークロード保護および Tetration エンドポイント可視性のライセンスタイプの予約数を更新します。

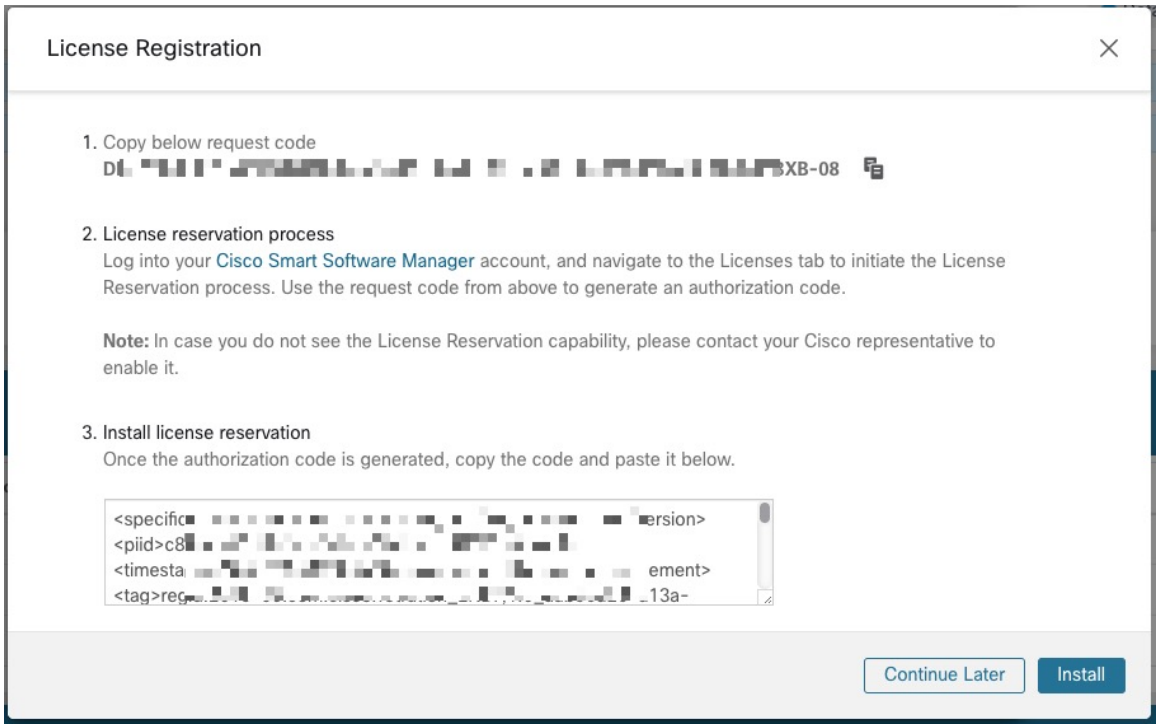
e) 選択したライセンスタイプと予約するライセンス数を確認し、[承認コードの生成（Generate Authorization Code）] をクリックします。

f) [クリップボードにコピー（Copy to Clipboard）] をクリックします。

g) [Close] をクリックしてウィザードを終了します。

ステップ 6 Cisco Secure Workload で、承認コードを入力し、[インストール（Install）] をクリックします。

図 *5 : Cisco Secure Workload* ライセンス予約ページ



CSSM ポータルでは、予約済みライセンスは [ライセンス（Licenses）] タブで確認できます。

（注）

- ライセンス予約プロセス中に、[後で続行（Continue Later）] を選択することもできます。予約プロセスを再開するには、[ライセンスの使用状況情報（License Usage Information）] ページで [予約の続行（Continue Reservation）] をクリックします。
- ライセンス予約をキャンセルする場合は、[ライセンスの使用状況情報（License Usage Information）] ページで [予約キャンセル（Cancel Reservation）] をクリックし、次のいずれかのアクションを実行します。
  - 承認コードを生成した場合、通常は、CSSM で使用するためにライセンスが予約されます。ライセンスを返却するには、CSSM から生成された承認コードを入力します。その後、Cisco Secure Workload によってリターンコードが生成されます。このリターンコードを CSSM に入力する必要があります。
  - CSSM にライセンスがない場合は、[CSSMにライセンスがありません（I do not have a license in CSSM）] タブを選択します。

特定ライセンス予約の更新

手順

ステップ 1 CSSM ポータルに移動し、次のアクションを実行します。

a) 予約済みの特定のライセンスの場合は、[アクション（Actions）] ドロップダウンメニューから [予約済みライセンスの更新（Update Reserved Licenses）] を選択します。

b) [特定のライセンスの予約（Reserve a specific license）] を選択し、必要な Tetration ワークロード保護および Tetration エンドポイント可視性のライセンスタイプの予約数を更新します。

c) [次へ（Next）] をクリックします。

d) 選択した予約を確認し、[承認コードの生成（Generate Authorization Code）] をクリックします。

e) [クリップボードにコピー（Copy to Clipboard）] をクリックします。

ステップ 2 Cisco Secure Workload で、次のアクションを実行して特定のライセンスの予約を更新します。

a) [ライセンスの使用状況情報（License Usage Information）] ページで、[予約の更新（Update Reservation）] をクリックします。

b) CSSM からの承認コードを入力します。

c) [確認コードの生成（Generate Confirmation Code）] をクリックします。

d) 生成された確認コードをコピーし、[次へ（Next）] をクリックします。

e) 生成された確認コードは、Cisco Secure Workload で後から復元することはできません。リターンコードを必ずコピーしてください。[コピー（Copy）]アイコンをクリックします。

ステップ 3 CSSM ポータルに移動し、次のアクションを実行して予約プロセスを完了します。

a) [製品インスタンス（Product Instances）] で、予約するライセンスを選択し、[アクション（Actions）] をクリックします。

b) [確認コードの入力（Enter Confirmation Code）] を選択します。

c) 確認コードを入力して、[OK] をクリックします。

ステップ 4 Cisco Secure Workload で、[完了（Finish）] をクリックしてウィザードを終了します。

[ライセンスの使用状況情報（License Usage Information）] ページを更新して、更新されたライセンス予約の詳細を表示します。

特定ライセンス予約の返却

手順

ステップ 1 Cisco Secure Workload の [ライセンスの使用状況情報（License Usage Information）] ページで、[登録解除（Deregister）] をクリックします。

ステップ 2 [リターンコードの生成（Generate return code）] をクリックします。

ステップ 3 生成されたリターンコードをコピーし、[次へ（Next）] をクリックします。

ステップ 4 後からコードを回復することはできないため、生成されたリターンコードを必ずコピーしてください。[コピー（Copy）] アイコンをクリックします。

ステップ 5 CSSM ポータルに移動し、次のアクションを実行して登録解除プロセスを完了します。

a) [製品インスタンス（Product Instances）] で、登録を解除するライセンスを選択し、[アクション（Actions）] をクリックします。

b) [削除（Remove）] を選択します。

c) リターンコードを入力し、[予約の削除（Remove Reservation）] をクリックします。

ステップ 6 Cisco Secure Workload で、[完了（Finish）] をクリックしてウィザードを終了します。

予約済みライセンスがスマートアカウントに返却され、Cisco Secure Workload は評価期間になり、未登録のコンプライアンス状態になります。

Cisco Secure Workload スマートライセンスの登録：CSSM オンプレミス

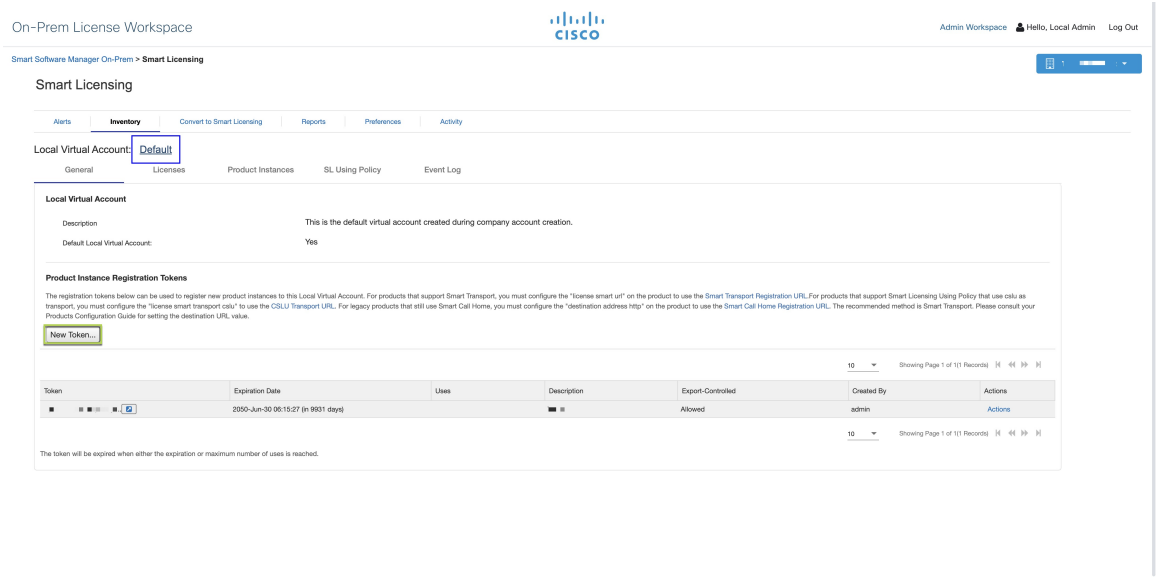
Smart Software Manager オンプレミスを使用することで、ライセンスの同期をスケジュールできます。また、Smart Software Manager ポータルを使用してスマートライセンス認証を手動で同期するオプションを使用できます。エアギャップネットワークで Cisco Secure Workload を登録するために、またはネットワークからの単一の接続を介してスマートライセンスを管理する場合に、CSSM オンプレミスを使用できます。

手順

ステップ 1 Smart Software Manager オンプレミスを設定します。詳細については、「Cisco Software Central」を参照してください。

ステップ 2 Smart Software Manager オンプレミスにログインし、仮想アカウントを選択します。[全般（General）] タブで、[新規トークン（New Token）] をクリックします。画面の指示に従って、トークンを作成します。

図 *6*:オンプレミス ライセンス ワークスペース



ステップ 3 Cisco Secure Workload で、**[管理（Manage）]** > **[サービスの設定（Service Settings）]** > **[ライセンス（Licenses）]** に移動し、[登録（Register）] をクリックします。

ステップ 4 [スマートライセンス（Smart Licenses）] タブで、Smart Software Manager オンプレミスから生成されたトークンを入力します。

ステップ 5 [スマートライセンス（Smart Licenses）] タブの [スマートトランスポート登録URL（Smart Transport Registration URL）] フィールドに、Smart Software Manager オンプレミスのスマートトランスポート登録 URL を入力します。

ステップ 6 （オプション）CSSM を使用してクラスタ UUID がすでに登録されている場合でも登録を続行するには、[強制登録（Force Register）] を有効にします。

ステップ 7 [Register] をクリックします。

CSSM オンプレミスを使用した Cisco Secure Workload の登録が開始されます。登録の更新されたステータスを表示するには、[ライセンスの使用状況情報（License Usage Information）] ページを更新します。ライセンスを割り当てた後に、Smart Software Manager オンプレミスと Smart Software Manager ポータルを同期します。

スマートライセンスの同期

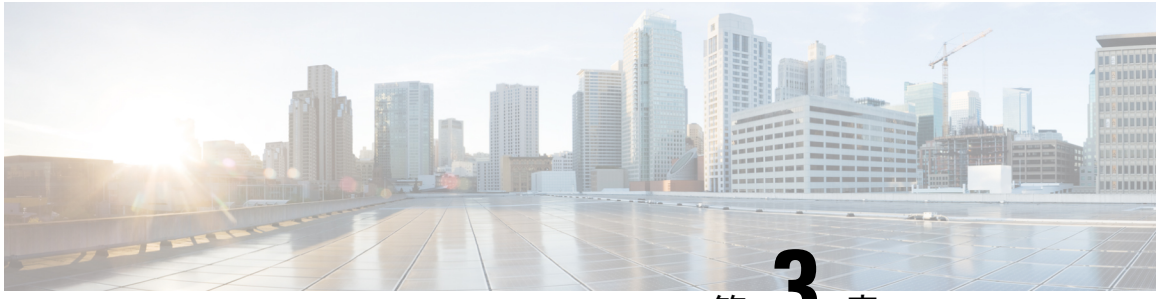
ライセンスおよびコンプライアンス情報は、24 時間ごとに Smart Software Manager ポータルと同期されます。ただし、クラスタがコンプライアンス違反であり、CSSM でライセンスが更新された場合は、CSSM ポータルに直接接続するか、次の手順で Smart Software Manager オンプレミスを介することで、ライセンスを手動で更新できます。

手順

ステップ 1 [ライセンスの使用状況情報（License Usage Information）] ページで、**[更新操作（Renew Operations）] > [認証の更新（Renew Authorization）]** をクリックします。

ステップ 2 確認するには、[はい（Yes）] をクリックします。

[ライセンスの使用状況情報（License Usage Information）] ページを更新すると、更新されたライセンスと、最後に CSSM と正常に同期された日時を表示できます。



第 3 章

ソフトウェアエージェント

Secure Workload ソフトウェアエージェントは、ワークロードにインストールする軽量のソフトウェアです。その目的は次のとおりです。

- システムで実行されているネットワークインターフェイスやアクティブなプロセスなどのホスト情報を収集します。
- ネットワークフロー情報をモニターおよび収集します。
- インストールされているホストにファイアウォールルールを設定して、（有効になっている場合）セキュリティポリシーを適用します。

エージェントは、インターフェイスアドレスが変更されると、Secure Workloadインベントリを自動的に更新します。

エンドユーザー（従業員）のコンピューターにエージェントをインストールする必要はありません。

- ソフトウェアエージェントの展開 （24 ページ）
- セキュリティの除外 （55 ページ）
- エージェントのサービスの管理 （59 ページ）
- エージェントによるポリシーの適用 （61 ページ）
- ソフトウェアエージェントの設定 （87 ページ）
- ワークロードプロファイルでの詳細なエージェントステータスの表示 （103 ページ）
- エージェントのリホーム （105 ページ）
- 適用が有効である場合のホスト IP アドレスの変更 （108 ページ）
- ソフトウェアエージェントのアップグレード （109 ページ）
- ソフトウェアエージェントの削除 （113 ページ）
- ワークロードエージェントにより収集されエクスポートされるデータ （117 ページ）
- 適用アラート （120 ページ）
- センサーアラート （126 ページ）
- ソフトウェアエージェントのトラブルシューティング （132 ページ）

ソフトウェアエージェントの展開

（注） 自動ロールマッピングを使用して LDAP/AD アカウントからダウンロードしたインストーラスクリプトは、ユーザーがログアウトするとすぐに失敗します。インストーラスクリプトにクラスタへの連続アクセスを許可するには、ユーザーに対して [ローカル認証を使用（Use Local Authentication）] を有効にすることを推奨します。

展開が成功すると、エージェントが実行されているホストに固有の一連のパラメータに基づき、Secure Workload クラスタによってエージェントに固有の ID が割り当てられます。ホスト名と BIOS UUID は設定の一部であるため、次の問題が発生する可能性があります。

1. BIOS UUID とホスト名を保持したまま仮想マシンを複製すると、登録に失敗します。登録の失敗は、VDI インスタントクローニングの実行時などにも発生する可能性があります。これは、Secure Workload クラスタに、同じパラメータセットを使用して登録されたソフトウェアエージェントがすでにあるためです。ユーザーは、OpenAPI を使用して登録済みのエージェントを削除できます。場合によっては、起動時に設定された重複 BIOS UUID が、遅延後に VMware によって変更されます。エージェントの登録は、Cisco Secure Workload サービスが再起動されない限り回復しません。
2. ホスト名を変更してホストを再起動すると、エージェントの新しい ID が生成される可能性があります。冗長または古いエージェントエントリは、一定時間が経過すると非アクティブとしてマークされます。ソフトウェアエージェントのトラブルシューティング（132 ページ）を参照してください。

サポートされているプラットフォームと要件

サポートされているプラットフォームとソフトウェアエージェントの追加要件については、以下を参照してください。

- ご使用のリリースのリリースノートについては、「Release Notes」を参照してください。
- Cisco Secure Workload Web ポータルのエージェント インストール ウィザード：左側のナビゲーションバーで **[管理（Manage）]**>**[エージェント（Agents）]** を選択し、[インストーラ（Installer）] タブをクリックします。インストール方法、プラットフォーム、およびエージェントタイプ（該当する場合）を選択して、サポートされているプラットフォームのバージョンを確認します。
- その他の依存関係については、「Support Matrix」を参照してください。すべての列が表示されていることを確認します。
- 各プラットフォームおよびエージェントタイプの追加要件は、以下の項で示されています。

優れた可視性と適用のための Linux エージェントのインストール

Linux エージェントのインストールの要件と前提条件

- 「サポートされているプラットフォームと要件」を参照してください。
- サービスをインストールして実行するには、ルート権限が必要です。
- エージェントおよびログファイルのストレージ要件：1 GB。
- ホストを監視しているセキュリティアプリケーションでセキュリティの除外を設定することにより、他のセキュリティアプリケーションがエージェントのインストールやエージェントのアクティビティをブロックしないようにします。「セキュリティの除外」を参照してください。
- エージェントがインストールされているホストには特別なユーザー **tet-sensor** が作成されることに注意してください。ホストで PAM/SELinux が設定されている場合は、tet-sensor プロセスの実行やコレクタへの接続など、tet-sensor ユーザーに適切な権限を付与する必要があります。代替のインストールディレクトリがあり、SELinux が設定されている場合は、代替のインストールディレクトリで実行が許可されていることを確認してください。
- エージェントが自動インストール（インストーラスクリプト）メソッドを使用してインストールされている場合は、unzip コマンドを使用できる必要があります。

Linux エージェントをインストールするためにサポートされている方法

優れた可視性と適用のために Linux エージェントをインストールする方法：

- インストーラを使用したエージェントの自動インストール
- 従来のパッケージインストーラを使用したエージェントの手動インストール
- エージェント スクリプト インストーラ方式を使用した Linux エージェントのインストール （25 ページ）
- エージェント イメージ インストーラ方式を使用した Linux エージェントのインストール （28 ページ）

エージェント スクリプト インストーラ方式を使用した Linux エージェントのインストール

インストーラスクリプトは、Linux プラットフォームで優れた可視性と適用のエージェントを展開する場合に推奨される方法です。

（注）
- インストールされた Linux エージェントは、優れた可視性と適用の両方をサポートしています。
- デフォルトでは、適用は無効になっています。適用を有効にするには、エージェント設定プロファイルの作成 （90 ページ）を参照してください。

スクリプトインストーラ方式を使用して Linux エージェントをインストールするには、次の手順を実行します。

手順

ステップ 1 エージェントのインストール方式に移動するために、次の手順を実行します。

- 初めてのユーザーの場合は、クイックスタートウィザードを起動し、[エージェントのインストール（Install Agents）] をクリックします。
- 左側のペインで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックし、[インストーラ（Installer）] タブを選択します。

ステップ 2 [エージェントスクリプトインストーラ（Agent Script Installer）] をクリックします。

ステップ 3 [プラットフォームの選択（Select Platform）] ドロップダウンメニューから、[Linux] を選択します。

サポートされている Linux プラットフォームを表示するには、[サポートされているプラットフォームの表示（Show Supported Platforms）] をクリックします。

ステップ 4 エージェントをインストールするテナントを選択します。

（注） Cisco Secure Workload SaaS クラスタでは、テナントを選択する必要はありません。

ステップ 5 （オプション）ワークロードにラベルを割り当てるには、ラベルキーを選択し、ラベル値を入力します。

インストールされたエージェントがホストで IP アドレスを報告すると、このホストによって報告された IP に割り当てられているアップロード済みの別の CMDB ラベルとともに、ここで選択されたインストーラ CMDB ラベルが新しい IP アドレスに自動的に割り当てられます。アップロード済みの CMDB ラベルとインストーラ CMDB ラベルの間で競合が発生した場合は、次のようになります。

- 正確な IP アドレスに割り当てられたラベルは、サブネットに割り当てられたラベルよりも優先されます。
- 正確な IP アドレスに割り当てられた既存のラベルは、インストーラの CMDB ラベルよりも優先されます。

ステップ 6 Cisco Secure Workload との通信に HTTP プロキシが必要な場合は、[はい（Yes）] を選択し、有効なプロキシ URL を入力します。

ステップ 7 [インストーラの有効期限（Installer expiration）] セクションで、利用可能なオプションから 1 つを選択します。

- [有効期限なし（No expiration）]：インストーラスクリプトは何回も使用できます。
- 1 回のみ：インストーラスクリプトは 1 回のみ使用できます。
- 時間制限：インストーラスクリプトを使用できる日数を設定できます。

• 展開数：インストーラスクリプトを使用できる回数を設定できます。

ステップ 8 [ダウンロード（Download）] をクリックし、ファイルをローカルディスクに保存します。

ステップ 9 Linux ホストでインストーラ シェル スクリプトをコピーし、`chmod u+x tetration_installer_default_sensor_linux.sh` コマンドを実行してスクリプトに実行権限を付与します。

（注） スクリプト名は、選択したエージェントのタイプと範囲によって異なる場合があります。

ステップ 10 エージェントをインストールするために、ルート権限で `./tetration_installer_default_sensor_linux.sh` コマンドを実行します。

（注） エージェントがテナントにすでにインストールされている場合は、インストールを続行できません。

スクリプトの使用方法の詳細で指定されているように、事前チェックを実行することを推奨します。

Linux インストーラスクリプトの使用方法の詳細：

```
bash tetration_linux_installer.sh [--pre-check] [--skip-pre-check=<option>] [--no-install]
 [--logfile=<filename>] [--proxy=<proxy_string>] [--no-proxy] [--help] [--version]
[--sensor-version=<version_info>] [--ls] [--file=<filename>] [--save=<filename>] [--new]
 [--reinstall] [--unpriv-user] [--force-upgrade] [--upgrade-local]
[--upgrade-by-uuid=<filename>] [--basedir=<basedir>] [--logbasedir=<logbdir>]
[--tmpdir=<tmp_dir>] [--visibility] [--golden-image]
  --pre-check: run pre-check only
  --skip-pre-check=<option>: skip pre-installation check by given option; Valid options
 include 'all', 'ipv6' and 'enforcement'; e.g.: '--skip-pre-check=all' will skip all
pre-installation checks; All pre-checks will be performed by default
  --no-install: will not download and install sensor package onto the system
  --logfile=<filename>: write the log to the file specified by <filename>
  --proxy=<proxy_string>: set the value of CL_HTTPS_PROXY, the string should be formatted
 as http://<proxy>:<port>
  --no-proxy: bypass system wide proxy; this flag will be ignored if --proxy flag was
provided
  --help: print this usage
  --version: print current script's version
  --sensor-version=<version_info>: select sensor's version; e.g.:
'--sensor-version=3.4.1.0'; will download the latest version by default if this flag was
 not provided
  --ls: list all available sensor versions for your system (will not list pre-3.1
packages); will not download any package
  --file=<filename>: provide local zip file to install sensor instead of downloading it
 from cluster
  --save=<filename>: download and save zip file as <filename>
  --new: remove any previous installed sensor; previous sensor identity has to be removed
 from cluster in order for the new registration to succeed
  --reinstall: reinstall sensor and retain the same identity with cluster; this flag has
 higher priority than --new
  --unpriv-user=<username>: use <username> for unpriv processes instead of tet-sensor
  --force-upgrade: force sensor upgrade to version given by --sensor-version flag; e.g.:
 '--sensor-version=3.4.1.0 --force-upgrade'; apply the latest version by default if
--sensor-version flag was not provided
  --upgrade-local: trigger local sensor upgrade to version given by --sensor-version
```

```
flag: e.g.: '--sensor-version=3.4.1.0 --upgrade-local'; apply the latest version by
default if --sensor-version flag was not provided
  --upgrade-by-uuid=<filename>: trigger sensor whose uuid is listed in <filename> upgrade
 to version given by --sensor-version flag; e.g.: '--sensor-version=3.4.1.0
--upgrade-by-uuid=/usr/local/tet/sensor_id'; apply the latest version by default if
--sensor-version flag was not provided
  --basedir=<base_dir>: instead of using /usr/local use <base_dir> to install agent. The
 full path will be <base_dir>/tetration
  --logbasedir=<log_base_dir>: instead of logging to /usr/local/tet/log use <log_base_dir>.
 The full path will be <log_base_dir>/tetration
  --tmpdir=<tmp_dir>: instead of using /tmp use <tmp_dir> as temp directory
  --visibility: install deep visibility agent only; --reinstall would overwrite this
flag if previous installed agent type was enforcer
  --golden-image: install Cisco Secure Workload Agent but do not start the Cisco Secure
 Workload Services; use to install Cisco Secure Workload Agent on Golden Images in VDI
environment or Template VM. On VDI/VM instance created from golden image with different
 host name, Cisco Secure Workload Services will work normally
```

（注）

- 現在、Ubuntu ではネイティブの .deb パッケージが使用されているため、新規インストールと再インストールを行うとこのパッケージタイプに切り替わります。以前のバージョンからのアップグレードには、引き続き .rpm パッケージが使用されます。
- Ubuntu .deb パッケージは、`/opt/cisco/tetration` にインストールされています。
- .deb パッケージの再配置がサポートされていないため、Ubuntu では –basedir オプションはサポートされていません。

エージェント イメージ インストーラ方式を使用した Linux エージェントのインストール

エージェントイメージ方式を使用してエージェントを手動でインストールする具体的な理由がない限り、（自動化されている）インストーラスクリプト方式を使用して Linux エージェントをインストールすることを推奨します。

前提条件：

SaaS クラスタの場合、および複数のテナントがあるオンプレミスクラスタでデフォルトテナントにインストールしない場合は、user.cfg ファイルで ACTIVATION_KEY と HTTPS_PROXY を設定する必要があります。詳細については、（手動インストールのみ）ユーザー構成ファイルの更新を参照してください。（手動インストールのみ）ユーザー構成ファイルの更新（52 ページ）

エージェントイメージ方式を使用して Linux エージェントをインストールするには、次の手順を実行します。

手順

ステップ 1 エージェントのインストール方式に移動するために、次の手順を実行します。

- 初めてのユーザーの場合は、クイックスタートウィザードを起動し、[エージェントのインストール（Install Agents）] をクリックします。

- 左側のペインで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックし、[インストーラ（Installer）] タブを選択します。

ステップ 2 [エージェントイメージインストーラ（Agent Image Installer）] をクリックします。

ステップ 3 [プラットフォーム（Platform）] フィールドに、Linux と入力します。

ステップ 4 必要なエージェントタイプとエージェントのバージョンを入力し、結果から必要なバージョンのエージェントをダウンロードします。

ステップ 5 RPM パッケージを展開するためにすべての Linux ホストにコピーします。

（注） エージェントがすでにホストにインストールされている場合は、エージェントを再インストールしないでください。エージェントをアップグレードするには、「ソフトウェアエージェントのアップグレード」を参照してください。

ステップ 6 プラットフォームに基づいて、ルート権限で RPM コマンドを実行します。

- RHEL/CentOS/Oracle プラットフォームの場合は、`rpm -ivh <rpm_filename>` コマンドを実行します。
- Ubuntu プラットフォームの場合：
  - 依存関係リストを取得し、すべての依存関係が満たされていることを確認するために、`rpm -qpR <rpm_filename>` コマンドを実行します。
  - `rpm -ivh \\--nodeps <rpm filename>` コマンドを実行して、“–nodeps” オプションを使用してエージェントをインストールします。

Linux エージェントのインストールの確認

手順

コマンド `sudo rpm -q tet-sensor` を実行します。
出力として単一のエントリがあることを確認します。これにより、Linux エージェントがホストにインストールされていることが確認されます。

出力例：`tet-sensor-3.1.1.50-1.el6.x86_64`

指定された出力は、プラットフォームとアーキテクチャによって異なる場合があります。

優れた可視性と適用のための Windows エージェントのインストール

Windows エージェントのインストールの要件と前提条件

- 「サポートされているプラットフォームと要件」を参照してください。
- インストールとサービス実行のための管理者権限が必要です。
- Windows 2008 R2 を実行しているワークロードでは、Npcap をインストールする必要があります。また、インストールされているエージェントのバージョンがバージョン 3.8 より前の場合も、Npcap をインストールする必要があります。Npcap ドライバがまだインストールされていない場合、推奨される Npcap バージョンが、サービスの開始後にエージェントによってサイレントにインストールされます。Npcap のバージョン情報を参照してください。
- エージェントおよびログファイルのストレージ要件：1 GB。
- 必要な Windows サービス：Windows ホストのセキュリティが強化されている場合や、Microsoft から出荷されたときのデフォルト構成から逸脱している場合は、エージェントのインストールを正常に行うために必要な一部の Windows サービスが無効になっている可能性があります。「必要な Windows サービス」を参照してください。
- ホストを監視しているセキュリティアプリケーションでセキュリティの除外を設定することにより、他のセキュリティアプリケーションがエージェントのインストールやエージェントのアクティビティをブロックしないようにします。「セキュリティの除外」を参照してください。

Windows エージェントをインストールするためにサポートされている方法

優れた可視性または適用の目的で Windows プラットフォームにエージェントをインストールするには、2 つの方法があります。

- インストーラを使用したエージェントの自動インストール
- 従来のパッケージインストーラを使用したエージェントの手動インストール

- エージェント スクリプト インストーラ方式を使用した Windows エージェントのインストール （30 ページ）
- エージェント イメージ インストーラ方式を使用した Windows エージェントのインストール （33 ページ）

ゴールデンイメージを使用したインストールも可能です。「VDI インスタンスまたは VM テンプレートへのエージェントの展開（Windows）」を参照してください。

エージェント スクリプト インストーラ方式を使用した Windows エージェントのインストール

インストーラスクリプトは、Windows プラットフォームで優れた可視性と適用のエージェントを展開する場合に推奨される方法です。

（注）
- インストールされた Windows エージェントは、優れた可視性と適用の両方をサポートしています。
- デフォルトでは、適用は無効になっています。適用を有効にするには、エージェント設定プロファイルの作成 （90 ページ）を参照してください。

スクリプトインストーラ方式を使用して Windows エージェントをインストールするには、次の手順を実行します。

手順

ステップ 1 エージェントのインストール方式に移動するために、次の手順を実行します。
- 初めてのユーザーの場合は、クイックスタートウィザードを起動し、[エージェントのインストール（Install Agents）] をクリックします。
- 左側のペインで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックし、[インストーラ（Installer）] タブを選択します。

ステップ 2 [エージェントスクリプトインストーラ（Agent Script Installer）] をクリックします。

ステップ 3 [プラットフォームの選択（Select Platform）] ドロップダウンメニューから、[Windows] を選択します。

サポートされている Windows プラットフォームを表示するには、[サポートされているプラットフォームの表示（Show Supported Platforms）] をクリックします。

ステップ 4 エージェントをインストールするテナントを選択します。

（注） Cisco Secure Workload SaaS クラスタでは、テナントを選択する必要はありません。

ステップ 5 （オプション）ワークロードにラベルを割り当てるには、ラベルキーを選択し、ラベル値を入力します。

インストールされたエージェントがホストで IP アドレスを報告すると、このホストによって報告された IP に割り当てられているアップロード済みの別の CMDB ラベルとともに、ここで選択されたインストーラ CMDB ラベルが新しい IP アドレスに自動的に割り当てられます。アップロード済みの CMDB ラベルとインストーラ CMDB ラベルの間で競合が発生した場合は、次のようになります。
- 正確な IP アドレスに割り当てられたラベルは、サブネットに割り当てられたラベルよりも優先されます。
- 正確な IP アドレスに割り当てられた既存のラベルは、インストーラの CMDB ラベルよりも優先されます。

ステップ 6 Cisco Secure Workload との通信に HTTP プロキシが必要な場合は、[はい（Yes）] を選択し、有効なプロキシ URL を入力します。

ステップ 7 [インストーラの有効期限（Installer expiration）] セクションで、利用可能なオプションから 1 つを選択します。

- [有効期限なし（No expiration）]：インストーラスクリプトは何回も使用できます。
- 1 回のみ：インストーラスクリプトは 1 回のみ使用できます。
- 時間制限：インストーラスクリプトを使用できる日数を設定できます。
- 展開数：インストーラスクリプトを使用できる回数を設定できます。

ステップ 8 [ダウンロード（Download）] をクリックし、ファイルをローカルディスクに保存します。

ステップ 9 インストーラの PowerShell スクリプトを展開用のすべての Windows ホストにコピーし、管理者権限でスクリプトを実行します。

（注） • システムの設定によっては、他のコマンドの前に、コマンド `Unblock-File` の実行が必要な場合があります。

スクリプトの使用方法の詳細で指定されているように、事前チェックを実行することを推奨します。

Windows インストーラスクリプトの使用方法の詳細：

```
# powershell -File tetration_windows_installer.ps1 [-preCheck] [-skipPreCheck <Option>]
 [-noInstall] [-logFile <FileName>] [-proxy <ProxyString>] [-noProxy] [-help] [-version]
 [-sensorVersion <VersionInfo>] [-ls] [-file <FileName>] [-save <FileName>] [-new]
[-reinstall] [
-npcap] [-forceUpgrade] [-upgradeLocal] [-upgradeByUUID <FileName>] [-visibility]
[-goldenImage] [-installFolder <Installation Path>]
  -preCheck: run pre-check only
  -skipPreCheck <Option>: skip pre-installation check by given option; Valid options
include 'all', 'ipv6' and 'enforcement'; e.g.: '-skipPreCheck all' will skip all
pre-installation checks; All pre-checks will be performed by default
  -noInstall: will not download and install sensor package onto the system
  -logFile <FileName>: write the log to the file specified by <FileName>
  -proxy <ProxyString>: set the value of HTTPS_PROXY, the string should be formatted as
 http://<proxy>:<port>
  -noProxy: bypass system wide proxy; this flag will be ignored if -proxy flag was
provided
  -help: print this usage
  -version: print current script's version
  -sensorVersion <VersionInfo>: select sensor's version; e.g.: '-sensorVersion
3.4.1.0.win64'; will download the latest version by default if this flag was not provided

  -ls: list all available sensor versions for your system (will not list pre-3.1 packages);
 will not download any package
  -file <FileName>: provide local zip file to install sensor instead of downloading it
from cluster
  -save <FileName>: downloaded and save zip file as <FileName>
  -new: remove any previous installed sensor; previous sensor identity has to be removed
 from cluster in order for the new registration to succeed
  -reinstall: reinstall sensor and retain the same identity with cluster; this flag has
 higher priority than -new
```

```
  -npcap: overwrite existing npcap
  -forceUpgrade: force sensor upgrade to version given by -sensorVersion flag; e.g.:
'-sensorVersion 3.4.1.0.win64 -forceUpgrade'; apply the latest version by default if
-sensorVersion flag was not provided
  -upgradeLocal: trigger local sensor upgrade to version given by -sensorVersion flag;
e.g.: '-sensorVersion 3.4.1.0.win64 -upgradeLocal'; apply the latest version by default
 if -sensorVersion flag was not provided
  -upgradeByUUID <FileName>: trigger sensor whose uuid is listed in <FileName> upgrade
to version given by -sensorVersion flag; e.g.: '-sensorVersion 3.4.1.0.win64 -upgradeByUUID
 "C:\\Program Files\\Cisco Tetration\\sensor_id"'; apply the latest version by default
if -sensorVersion flag was not provided
  -visibility: install deep visibility agent only; -reinstall would overwrite this flag
 if previous installed agent type was enforcer
  -goldenImage: install Cisco Secure Workload Agent but do not start the Cisco Secure
Workload Services; use to install Cisco Secure Workload Agent on Golden Images in VDI
environment or Template VM. On VDI/VM instance created from golden image with different
 host name, Cisco Secure Workload Services will work normally
  -installFolder: install Cisco Secure Workload Agent in a custom folder specified by
-installFolder e.g.: '-installFolder "c:\\custom sensor path"'; default path is "C:\Program
 Files\Cisco Tetration"
```

エージェント イメージ インストーラ方式を使用した Windows エージェントのインストール

エージェントイメージ方式を使用してエージェントを手動でインストールする具体的な理由がない限り、（自動化されている）インストーラスクリプト方式を使用して Windows エージェントをインストールすることを推奨します。

（注） ホストで既存のエージェントがすでに実行されている場合は、古いMSIエージェントバージョンを手動で展開しないでください。

パッケージ内には次のサイト関連ファイルがあります。

- **ca.cert**（必須）：センサー通信用の CA 証明書。
- **enforcer.cfg**（適用センサーをインストールする場合にのみ必須）：適用エンドポイントの構成が含まれています。
- **sensor_config**（必須）：優れた可視性センサーの構成。
- **sensor_type**：センサーのタイプ（適用または優れた可視性）。
- **site.cfg**（必須）：グローバル サイト エンドポイント構成。
- **user.cfg**（SaaS の場合は必須）：センサー アクティベーション キーとプロキシ構成。

前提条件：

SaaSクラスタの場合、および複数のテナントがあるオンプレミスクラスタでデフォルトテナントにインストールしない場合は、user.cfg ファイルで ACTIVATION_KEY と HTTPS_PROXY を設定する必要があります。詳細については、（手動インストールのみ）ユーザー構成ファイルの更新を参照してください。（手動インストールのみ）ユーザー構成ファイルの更新（52ページ）

エージェントイメージ方式を使用して Windows エージェントをインストールするには、次の手順を実行します。

手順

ステップ 1 エージェントのインストール方式に移動するために、次の手順を実行します。

- 初めてのユーザーの場合は、クイックスタートウィザードを起動し、[エージェントのインストール（Install Agents）] をクリックします。
- 左側のペインで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックし、[インストーラ（Installer）] タブを選択します。

ステップ 2 [エージェントイメージインストーラ（Agent Image Installer）] をクリックします。

ステップ 3 [プラットフォーム（Platform）] フィールドに、Windows と入力します。

ステップ 4 必要なエージェントタイプとエージェントのバージョンを入力し、結果から必要なバージョンのエージェントをダウンロードします。

ステップ 5 tet-win-sensor<version>.win64-<clustername>.zip ファイルを、展開用のすべての Windows ホストにコピーします。

ステップ 6 管理者権限で ZIP ファイルを解凍します。

ステップ 7 解凍したフォルダで、コマンド `msiexec.exe /i TetrationAgentInstaller.msi` を実行してエージェントをインストールします。

さらに、MSI インストーラでは次のオプションを使用できます。

表 *2: MSI* インストーラで使用可能なオプション

オプション	説明
agenttype=<AgentType>	AgentType は、適用が必要かどうかに応じて、sensor または enforcer のいずれかにする必要があります。デフォルトでは、sensor_type ファイルの内容が同じフォルダ内のインストーラによってチェックされ、渡されたパラメータが上書きされます。ただし、エージェントが /quiet モードでインストールされている場合、このオプションは必須です。

オプション	説明
overwritenpcap=yes	Windows 2008 R2 の場合、デフォルトでは、Npcap がすでに存在する場合、エージェントは Npcap のアップグレードを試行しません。このパラメータを渡すと、既存の Npcap がアップグレードされます。このオプションが使用されている場合、後続のエージェントの自動アップグレードでも、Npcap がサポートされている新しいバージョンにアップグレードされます。
nostart=yes	エージェントサービス（tetsensor および tetenforcer）が自動的に開始されないようにするために、VDI 環境または VM テンプレートのゴールデンイメージにエージェントをインストールする場合、このパラメータを渡す必要があります。異なるホスト名を持つゴールデンイメージから作成された VDI/VM インスタンスでは、これらのサービスは期待どおりに自動的に開始されます。
installfolder=<FullPathCustomFolder>	インストールコマンドの最後にこのパラメータを使用して、カスタムフォルダにエージェントをインストールします。
serviceuser=<Service UserName>	インストールコマンドの最後にこのパラメータを使用して、サービスユーザーを構成します。デフォルトのサービスユーザーは「LocalSystem」です。 ローカルユーザーの場合、serviceuser=.\<Service UserName> ドメインユーザーの場合、serviceuser=<domain_name>\<samaccount name> サービスユーザーはローカルの管理者権限を持つ必要があります。
servicepassword=<Service UserPassword>	インストールコマンドの最後にこのパラメータを使用して、サービスユーザーのパスワードを構成します。パスワードはプレーンテキスト形式である必要があります。
proxy=”<proxy_address>”	このパラメータを使用して、Cisco Secure Workload クラスタにアクセスするための HTTPS プロキシを設定します。

オプション	説明
activationkey=<activation Key>	デフォルトテナント配下にエージェントがインストールされていない場合は、このパラメータでテナントを指定します。



（注）

- 手動インストール中に activation key および proxy オプションを使用する場合、user.cfg を手動で構成する必要はありません。
- Npcap がホストにインストールされていない場合は、Cisco Secure Workload サービスの tetsensor によって Npcap が自動的にインストールされます。
- Windows 2008 R2 以外の Windows OS の場合、バージョン 3.8 にアップグレードすると、インストールされている Npcap が Windows エージェントによって自動的にアンインストールされます。
- エージェントがすでにホストにインストールされている場合は、エージェントを再インストールしないでください。エージェントをアップグレードするには、「ソフトウェアエージェントのアップグレード」を参照してください。

Windows エージェントのインストールの確認

手順

ステップ 1 フォルダ `C:\\Program Files\\Cisco Tetration`（またはカスタムフォルダ）が存在することを確認します。

ステップ 2 TetSensor サービス（優れた可視性エージェント向け）が存在し、実行状態であることを確認します。管理者権限で `cmd.exe` コマンドを実行します。

`sc query tetsensor` コマンドを実行します。

状態が **Running** であることを確認します。

`sc query tetsensor` コマンドを実行します。

DISPLAY-NAME が **Cisco Secure Workload Deep Visibility** であるかを確認します。

または

services.msc コマンドを実行します。

Cisco Secure Workload Deep Visibility という名前を見つけます。

状態が **Running** であることを確認します。

ステップ 3 TetEnforcerサービス（適用エージェント向け）が存在し、実行状態であることを確認します。

管理者権限で `cmd.exe` コマンドを実行します。

`sc query tetenforcer` コマンドを実行します。

状態が **Running** であることを確認します。

`sc qc tetenforcer` コマンドを実行します。

DISPLAY-NAME が **Cisco Secure Workload Enforcement** であるかを確認します。

または

`services.msc` コマンドを実行します。

Cisco Secure Workload Enforcement という名前を見つけます。

状態が **Running** であることを確認します。

設定されたサービス ユーザー コンテキストの Windows エージェントの確認

1. TetSensor（詳細可視性のため）と TetEnforcer（適用のため）の両サービスが設定されたサービスユーザーコンテキストで実行されていることを確認します。TetSensor と TetEnforcer は、同じサービス ユーザー コンテキストで実行されます。

   管理者権限で `cmd.exe` コマンドを実行します。

   `sc query tetsensor` コマンドを実行します。

   SERVICE_START_NAME **<configured service user>** を確認します。

   `sc qc tetenforcer` コマンドを実行します。

   SERVICE_START_NAME **<configured service user>** を確認します。

   または

   `services.msc` コマンドを実行します。

   Cisco Secure Workload Deep Visibility という名前を見つけます。

   <configured service user> の **Log On As** を確認します。

   Cisco Secure Workload Enforcement という名前を検索します。

   <configured service user> の **Log On As** を確認します。

   または

   `tasklist /v | find /i "tet"` コマンドを実行します。

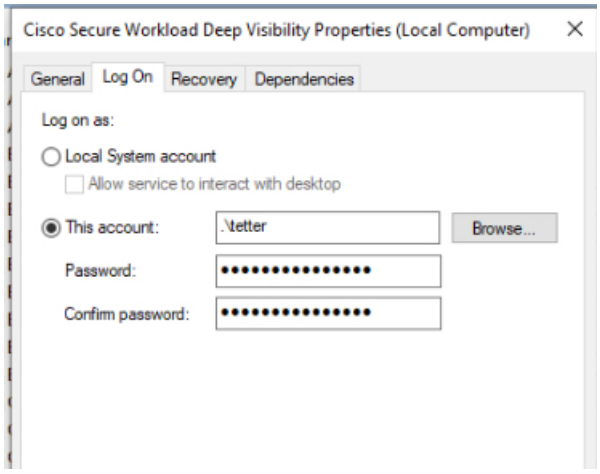
   実行中のプロセスのユーザーコンテキストを確認します（5 列目）。

サービスアカウントの変更

Windowsエージェントのインストール後にサービスアカウントを変更するには、次のいずれかの方法を使用して、既存の優れた可視性および適用のサービスを変更します。

- **services.msc** を使用します。

図 *7 : services.msc* アカウントに基づくサービスアカウントの変更



- 任意のサードパーティ アプリケーションを使用して、サービスを構成します。
- 次のコマンドを使用します。

1. 管理者として cmd を実行します。
2. 次のコマンドを実行して、サービスアカウント名を使用してサービスを変更します。
    1. `sc config tetsensor obj=<service user name>パスワード=<password>`
    2. `sc config tetenforcer obj= <service user name> password= <password>`
3. 次のコマンドを実行して構成を確認します。
    1. `sc qc tetsensor`
    2. `sc qc tetenforcer`
4. 次のコマンドを実行して、tetsensor と tetenforcer サービスを再開します。
    1. `sc stop tetsensor / tetenforcer`
    2. `sc start tetsensor / tetenforcer`

VDI インスタンスまたは VM テンプレートへのエージェントの展開（Windows）

デフォルトでは、エージェントのインストール後にエージェントサービスが自動的に開始されます。ゴールデンイメージにインストールする場合は、インストーラフラグを使用して、エー

ジェントサービスが開始されないようにする必要があります。インスタンスがゴールデンイメージから複製されると、エージェントサービスは予想どおりに自動的に開始されます。

同様に、Npcapは通常、エージェントのインストール後に自動的にインストールされます（エージェントがまだ存在していない場合）。Npcapはゴールデンイメージに自動的にインストールされませんが、必要に応じて、ゴールデンイメージから複製されたVMインスタンスに自動的にインストールされます。詳細については、「Windows エージェントインストーラと Npcap：Windows 2008 R2 の場合」を参照してください。

エージェントではゴールデンVMにNpcapはインストールされませんが、必要に応じて、ゴールデンイメージから複製されたVMインスタンスに自動的にインストールされます。詳細については、「Windows エージェントインストーラと Npcap：Windows 2008 R2 の場合」を参照してください。

VDI 環境または VM テンプレートのゴールデンイメージにエージェントをインストールする

手順

ステップ 1 MSI インストーラまたは PowerShell インストーラスクリプトを使用して、VDI 環境または VM テンプレートのゴールデンイメージにエージェントをインストールします。

nostart=yes を指定した MSI インストーラーを使用

- 詳細については、「エージェント イメージ インストーラ方式を使用した Windows エージェントのインストール （33 ページ）」を参照してください。
- msiexec.exe /<MSI installer> nostart=”yes” /quiet /norestart /l*v <installer_log_file>

または

-goldenImage フラグを指定した PowerShell インストーラを使用

- 詳細については、「エージェント スクリプト インストーラ方式を使用した Windows エージェントのインストール （30 ページ）」を参照してください。

ステップ 2 フォルダ `C:\\Program Files\\Cisco Tetration`（またはカスタムフォルダ）が存在することを確認します。

ステップ 3 TetSensor サービス（優れた可視性エージェント向け）が存在し、停止していることを確認します。

管理者権限で `cmd.exe` コマンドを実行します。

`sc query tetsensor` コマンドを実行します。

状態が **Stopped** であることを確認します。

ステップ 4 TetEnforcer サービス（適用エージェント向け）が存在し、停止していることを確認します。

sc query tetenforcer コマンドを実行します。

状態が Stopped であることを確認します。

ステップ 5 これで VM テンプレートが設定されました。

ステップ 6 VM テンプレートをシャットダウンします。

新規 VDI インスタンス VM の作成

手順

ステップ 1 VM テンプレートを複製して、新規 VDI インスタンス VM を作成します。

ステップ 2 VDI インスタンス VM を再起動します。

ステップ 3 VDI インスタンス VM を再起動した後、TetSensor（詳細可視性のため）と TetEnforcer（適用のため）の両サービスが設定されたサービスコンテキストで実行されていることを確認します。「Windows エージェントのインストールの確認」を参照してください。

ステップ 4 VDI インスタンス VM で、NPCAP ドライバがインストールされ、実行されていることを確認します。

管理者権限でコマンド `cmd.exe` を実行します。

コマンド `sc query npcap` を実行します。

[実行中（Running）] の状態を確認します。

ステップ 5 VDI インスタンス VM で、有効な sensor_id を使用してエージェントが登録されていることを確認します。

- インストールフォルダの sensor_id ファイルを確認します。
- sensor_id が「uuid」で始まる場合、それは有効な sensor_id ではありません。
- エージェントの登録に失敗したにもかかわらず、Secure Workload Web インターフェイスにエージェントが登録済みと表示されている場合：
- OpenAPI を使用してエージェントを削除してください。「ソフトウェアエージェントの展開」の下の注記を参照してください。

（注）
- ゴールデンイメージまたはVMテンプレートのホスト名は変更しないでください。
- エージェントのインストール後にゴールデンイメージまたはVMテンプレートを再起動すると、再起動後にSecure Workloadサービスの実行が開始されます。
- VDI インスタンス VM がネットワークフローのレポートに失敗する場合は、「ネットワークフローを報告しない VDI インスタンス VM 」を参照してください。

Windows エージェントインストーラと Npcap：Windows 2008 R2 の場合

1. サポートされている Npcap バージョンについては、https://www.cisco.com/go/secure-workload/requirements/agents のサポートマトリックスを参照してください。
2. インストール：

   Npcap がインストールされていない場合、エージェントはサービスの開始から 10 秒後にサポートされているバージョンをインストールします。ユーザーが Npcap をインストールしていても、サポートされているバージョンより古い場合、Npcap はアップグレードされません。Npcap を自分でアップグレード/アンインストールするか、オプション **overwritenpcap=yes** を指定してエージェントインストーラを実行するか、または **-npcap** を指定してインストーラスクリプトを実行して、サポートされている Npcap バージョンを取得してください。Npcap ドライバがアプリケーションで使用されている場合、エージェントは後で Npcap のアップグレードを試みます。
3. アップグレード：

   Npcap が Windows エージェントによってインストールされ、そのバージョンがサポートされているバージョンより古い場合、Npcap はサービスの開始から 10 秒後にサポートされているバージョンにアップグレードされます。Npcap ドライバがアプリケーションで使用されている場合、エージェントは後で Npcap のアップグレードを試みます。Npcap が Windows エージェントによってインストールされていない場合、Npcap はアップグレードされません。
4. アンインストール：

   Npcap が Windows エージェントによってインストールされている場合、Npcap は Windows エージェントによってアンインストールされます。Npcap がユーザーによってインストールされていても、**overwritenpcap=yes** でエージェントインストーラによってアップグレードされた場合、アンインストールされません。Npcap ドライバがアプリケーションで使用されている場合、エージェントは Npcap をアンインストールしません。

Windows エージェント フロー キャプチャ：Windows 2008 R2 を除くすべての Windows OS の場合

最新バージョンの Windows エージェントからは、エージェントは、ndiscap.sys（Microsoft 組み込み）ドライバと、Windows を使用したイベントトレース（ETW）フレームワークを使用して、ネットワークフローをキャプチャします。

最新バージョンへのアップグレード中に、以下のことが行われます。

- エージェントが npcap.sys から ndiscap.sys に切り替わります。
- エージェントインストーラが次の場合に Npcap をアンインストールします。
  - Npcap がエージェントによってインストールされている
  - Npcap が使用されていない
  - OS バージョンが Windows 2008 R2 ではない

エージェントサービスが開始されると、エージェントが、ETW セッション、CSW_MonNet、および CSW_MonDns（DNS データ用）を作成し、ネットワークフローのキャプチャを開始します。

（注） Windows 2012 では、DNS データのネットワークパケットが解析されます。

優れた可視性と適用のための AIX エージェントのインストール

（注） プロセスツリー、パッケージ（CVE）、およびフォレンジックイベントレポート機能は、AIX では使用できません。さらに、これらの機能の一部は、OS の制限により、サポートされているプラットフォームの特定のマイナーリリースでは利用できない場合があります。

AIX エージェントのインストールの要件と前提条件

- 「サポートされているプラットフォームと要件」を参照してください。
- 優れた可視性のための追加要件：
  - サービスをインストールして実行するには、ルート権限が必要です。
  - エージェントおよびログファイルのストレージ要件：500 MB。
  - ホストを監視しているすべてのセキュリティアプリケーションでセキュリティの除外を構成することにより、他のセキュリティアプリケーションがエージェントのインストールまたはエージェントのアクティビティをブロックしないようにします。「セキュリティの除外」を参照してください。
  - AIX は、20 個のネットデバイスのフローキャプチャのみをサポートします（バージョンが AIX 7.1 TL3 SP4 以前の場合は 6 個のネットデバイス）。優れた可視性エージェントは、最大 16 個のネットワークデバイスからキャプチャを行い、他の 4 個のキャプチャセッションを一般的なシステム用途（tcpdump など）に排他的に使用できるようにします。
  - 優れた可視性エージェントは、この動作を確実にするために次のことを行います。
    - エージェントは、エージェントディレクトリ（/opt/cisco/tetration/chroot/dev/bpf0 - /opt/cisco/tetration/chroot/dev/bpf15）の下に 16 個の bpf デバイスノードを作成します。
    - bpf を使用する tcpdump およびその他のシステムツールは、未使用のノード（!EBUSY）が見つかるまで、システムデバイスノード（/dev/bpf0-/dev/bpf19）をスキャンします。

- エージェントが作成した bpf ノードとシステム bpf ノードは同じメジャー/マイナーを共有し、各メジャー/マイナーは 1 つのインスタンス（tcpdump またはエージェント）によってのみ開かれます。
- エージェントはシステムデバイスノードにアクセスせず、tcpdump のようにそれらを作成しません（tcpdump-D は /dev/bpf0. . . /dev/bpf19 が存在しない場合これらを作成します）。

- システムで iptrace を実行すると、特定のシナリオで tcpdump および優れた可視性エージェントからのフローキャプチャが防止されます。これは設計上の既知の問題であり、IBM に確認する必要があります。
  - エージェントをインストールする前にこのシナリオが存在するかどうかを確認するには、tcpdump を実行します。**tcpdump: BIOCSETIF: en0: File exists** のようなエラーメッセージが表示される場合、iptrace はフローキャプチャをブロックしています。iptrace を停止すると、問題が解決します。

- AIX では、すべての優れた可視性機能がサポートされているわけではありません。例えば、パッケージとプロセスのアカウンティングはサポートされていません。

- ポリシーの適用に関する追加要件：
  - IP セキュリティフィルタが有効になっている場合（例：smitty ipsec4）、事前チェックでエージェントのインストールが失敗します。エージェントをインストールする前に、IP セキュリティフィルタを無効にすることを推奨します。
  - IP セキュリティが有効になっている場合、Cisco Secure Workload エンフォーサエージェントの実行中にエラーとして報告され、適用エージェントは適用を停止します。適用エージェントの実行中に IP セキュリティフィルタを安全に無効にするには、サポートにお問い合わせください。

エージェントスクリプトインストーラ方式を使用したAIXエージェントのインストール

優れた可視性または適用の AIX エージェントは、エージェント スクリプト インストール方式を使用しないとインストールできません。

（注）
- インストールされた AIX エージェントは、優れた可視性と適用の両方をサポートしています。
- デフォルトでは、適用は無効になっています。適用を有効にするには、エージェント設定プロファイルの作成 （90 ページ）を参照してください。

AIX エージェントをインストールするには、次の手順を実行します。

手順

ステップ 1 エージェントのインストール方式に移動するために、次の手順を実行します。

- 初めてのユーザーの場合は、クイックスタートウィザードを起動し、[エージェントのインストール（Install Agents）] をクリックします。
- 左側のペインで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックし、[インストーラ（Installer）] タブを選択します。

ステップ 2 [エージェントスクリプトインストーラ（Agent Script Installer）] をクリックします。

ステップ 3 [プラットフォームの選択（Select Platform）] ドロップダウンメニューから、[AIX] を選択します。

サポートされている AIX プラットフォームを表示するには、[サポートされているプラットフォームの表示（Show Supported Platforms）] をクリックします。

ステップ 4 エージェントをインストールするテナントを選択します。

（注） Cisco Secure Workload SaaS クラスタでは、テナントを選択する必要はありません。

ステップ 5 （オプション）ワークロードにラベルを割り当てるには、ラベルキーを選択し、ラベル値を入力します。

インストールされたエージェントがホストで IP アドレスを報告すると、このホストによって報告された IP に割り当てられているアップロード済みの別の CMDB ラベルとともに、ここで選択されたインストーラ CMDB ラベルが新しい IP アドレスに自動的に割り当てられます。アップロード済みの CMDB ラベルとインストーラ CMDB ラベルの間で競合が発生した場合は、次のようになります。

- 正確な IP アドレスに割り当てられたラベルは、サブネットに割り当てられたラベルよりも優先されます。
- 正確な IP アドレスに割り当てられた既存のラベルは、インストーラの CMDB ラベルよりも優先されます。

ステップ 6 Cisco Secure Workload との通信に HTTP プロキシが必要な場合は、[はい（Yes）] を選択し、有効なプロキシ URL を入力します。

ステップ 7 [インストーラの有効期限（Installer expiration）] セクションで、利用可能なオプションから 1 つを選択します。

- [有効期限なし（No expiration）]：インストーラスクリプトは何回も使用できます。
- 1 回のみ：インストーラスクリプトは 1 回のみ使用できます。
- 時間制限：インストーラスクリプトを使用できる日数を設定できます。
- 展開数：インストーラスクリプトを使用できる回数を設定できます。

ステップ 8 [ダウンロード（Download）] をクリックし、ファイルをローカルディスクに保存します。

ステップ 9 インストーラ シェル スクリプトを、展開するためにすべての AIX ホストにコピーします。

ステップ 10 スクリプトに実行権限を付与するために、`chmod u+x tetration_installer_default_sensor_aix.sh` コマンドを実行します。

（注） スクリプト名は、エージェントのタイプと範囲によって異なる場合があります。

ステップ 11 エージェントをインストールするために、ルート権限で `./tetration_installer_default_sensor_aix.sh` コマンドを実行します。

（注） エージェントがホストにすでにインストールされている場合は、インストールを続行できません。

スクリプトの使用方法の詳細で指定されているように、事前チェックを実行することを推奨します。

AIX インストーラスクリプトの使用方法の詳細：

```
ksh tetration_installer_default_enforcer_aix.sh [--pre-check] [--pre-check-user]
[--skip-pre-check=<option>] [--no-install] [--logfile=<filename>] [--proxy=<proxy_string>]
 [--no-proxy] [--help] [--version] [--sensor-version=<version_info>] [--ls]
[--file=<filename>] [--osversion=<osversion>] [--save=<filename>] [--new] [--reinstall]
 [--unpriv-user] [--libs=<libs.zip|tar.Z>] [--force-upgrade] [--upgrade-local]
[--upgrade-by-uuid=<filename>] [--logbasedir=<logbdir>] [--tmpdir=<tmp_dir>] [--visibility]
 [--golden-image]
  --pre-check: run pre-check only
  --pre-check-user: provide alternative to nobody user for pre-check su support
  --skip-pre-check=<option>: skip pre-installation check by given option; Valid options
 include 'all', 'ipv6' and 'enforcement'; e.g.: '--skip-pre-check=all' will skip all
pre-installation checks; All pre-checks will be performed by default
  --no-install: will not download and install sensor package onto the system
  --logfile=<filename>: write the log to the file specified by <filename>
  --proxy=<proxy_string>: set the value of HTTPS_PROXY, the string should be formatted
as http://<proxy>:<port>
  --no-proxy: bypass system wide proxy; this flag will be ignored if --proxy flag was
provided
  --help: print this usage
  --version: print current script's version
  --sensor-version=<version_info>: select sensor's version; e.g.:
'--sensor-version=3.4.1.0'; will download the latest version by default if this flag was
 not provided
  --ls: list all available sensor versions for your system (will not list pre-3.3
packages); will not download any package
  --file=<filename>: provide local zip file to install sensor instead of downloading it
 from cluster
  --osversion=<osversion>: specify osversion for --save flag;
  --save=<filename>: download and save zip file as <filename>; will download package for
 osversion given by --osversion flag; e.g.: '--save=myimage.aix72.tar.Z --osversion=7.2'

  --new: remove any previous installed sensor; previous sensor identity has to be removed
 from cluster in order for the new registration to succeed
  --reinstall: reinstall sensor and retain the same identity with cluster; this flag has
 higher priority than --new
  --unpriv-user=<username>: use <username> for unpriv processes instead of tet-snsr
  --libs=<libs.zip|tar.Z>: install provided libs to be used by agents
  --force-upgrade: force sensor upgrade to version given by --sensor-version flag; e.g.:
 '--sensor-version=3.4.1.0 --force-upgrade'; apply the latest version by default if
```

```
--sensor-version flag was not provided
  --upgrade-local: trigger local sensor upgrade to version given by --sensor-version
flag: e.g.: '--sensor-version=3.4.1.0 --upgrade-local'; apply the latest version by
default if --sensor-version flag was not provided
  --upgrade-by-uuid=<filename>: trigger sensor whose uuid is listed in <filename> upgrade
 to version given by --sensor-version flag; e.g.: '--sensor-version=3.4.1.0
--upgrade-by-uuid=/usr/local/tet/sensor_id'; apply the latest version by default if
--sensor-version flag was not provided
  --logbasedir=<log_base_dir>: instead of logging to /opt/cisco/tetration/log use
<log_base_dir>. The full path will be <log_base_dir>/tetration
  --tmpdir=<tmp_dir>: instead of using /tmp use <tmp_dir> as temp directory
  --visibility: install deep visibility agent only; --reinstall would overwrite this
flag if previous installed agent type was enforcer
  --golden-image: install Cisco Secure Workload Agent but do not start the Cisco Secure
 Workload Services; use to install Cisco Secure Workload Agent on Golden Images in VDI
environment or Template VM. On VDI/VM instance created from golden image with different
 host name, Cisco Secure Workload Services will work normally
```

AIX エージェントのインストールの確認

手順

コマンド `lslpp -c -l tet-sensor.rte` を実行し、以下のように 1 つのエントリがあることを確認します。

（注）　具体的な出力データは、バージョンによって異なる場合があります。

`$ sudo lslpp -c -l tet-sensor.rte` /usr/lib/objrepos:tet-sensor.rte:3.4.1.19::COMMITTED:I:TET tet sensor package:

```
$ sudo lssrc -s tet-sensor
```

Subsystem Group PID Status tet-sensor 1234567 active

```
$ sudo lssrc -s tet-enforcer
```

Subsystem Group PID Status tet-enforcer 7654321 active

優れた可視性と適用のための Kubernetes または OpenShift エージェントのインストール

要件および前提条件

Kubernetes 1.[16-22]

- RHEL：7.[0-9]（x86_64 アーキテクチャのみ）
- CentOS：7.[0-8]（x86_64 アーキテクチャのみ）
- Oracle Linux：7.[0-8]（x86_64 アーキテクチャのみ）

- Ubuntu：16.04、18.04、20.04（x86_64 アーキテクチャのみ）
- SUSE Linux Enterprise Server：12sp[0-5]（x86_64 アーキテクチャのみ）
- Amazon Linux 2（x86_64 アーキテクチャのみ）

Openshift 4.[5-9]

- Red Hat Enterprise Linux CoreOS：4.[5-9]（x86_64 アーキテクチャのみ）

コンテナ ランタイム

- Docker
- CRI-O
- containerd（>= 1.5.x）



（注） containerd ランタイムでは、config_path が設定されていない場合は、config.toml（デフォルトの場所：/etc/containerd/config.toml）を次のように変更します。

[plugins."io.containerd.grpc.v1.cri".registry] config_path = "/etc/containerd/certs.d"

containerd デーモンを再起動します。

追加の要件

- インストールスクリプトには、クラスタノードで特権エージェントポッドを起動するための Kubernetes または OpenShift 管理者のログイン情報が必要です。
- Cisco Secure Workload エンティティは、'tetration' という名前の名前空間に作成されます。
- ノードまたはポッドのセキュリティポリシーは、特権モードのポッドを許可する必要があります。
- busybox:1.33 イメージは、事前にインストールされているか、Docker Hub からダウンロード可能である必要があります。
- Kubernetes または OpenShift コントロールプレーンノードで実行するには、-toleration フラグを使用して、Secure Workload ポッドの容認を渡すことができます。これは通常、ポッドがコントロールプレーンノードで実行されないようにする NoSchedule 容認です。

ポリシー適用の要件

コンテナ オーケストレーション プラットフォームでポリシーを適用するエージェントは、RHEL 7.[0-9]、CentOS 7.[0-8]、または Ubuntu 16.04/18.04/20.04 ノードでサポートされます。

IPVS ベースの kube-proxy モードは、OpenShift ではサポートされません。

これらのエージェントは、[ルールの保持（Preserve Rules）] オプションを有効にして設定する必要があります。「エージェント設定プロファイルの作成」を参照してください。

適用が適切に機能するためには、インストールされている CNI プラグインが次の条件を満たす必要があります。

- すべてのノードとポッド間にフラットなアドレス空間（IP ネットワーク）を提供すること。クラスタ内通信のためにソースポッド IP をマスカレードするネットワークプラグインはサポートされていません。
- Secure Workload 適用エージェントが使用する Linux iptables ルールまたはマークに干渉しないこと（マークビット 21 および 20 は、NodePort サービスのトラフィックを許可および拒否するために使用されます）

次の CNI プラグインは、上記の要件を満たすことがテストされています。

- 次の Felix 設定を持つ Calico (3.13)：*(ChainInsertMode: Append, Ipta- blesRefreshInterval: 0)* または *(ChainInsertMode: Insert, IptablesFilterAllowAction: Return, IptablesMangleAllowAction: Return, IptablesRefreshInterval: 0)*。その他のオプションは、そのデフォルト値を使用します。

これらのオプションの設定に関する詳細については、Felix 設定リファレンスを参照してください。

エージェント スクリプト インストーラ方式を使用した Kubernetes または OpenShift エージェントのインストール

（注） エージェント スクリプト インストーラ方式では、後から含まれるノードにエージェントが自動的にインストールされます。

手順

ステップ 1 エージェントのインストール方式に移動するために、次の手順を実行します。

- 初めてのユーザーの場合は、クイックスタートウィザードを起動し、[エージェントのインストール（Install Agents）] をクリックします。
- 左側のペインで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックし、[インストーラ（Installer）] タブを選択します。

ステップ 2 [エージェントスクリプトインストーラ（Agent Script Installer）] をクリックします。

ステップ 3 [プラットフォームの選択（Select Platform）] ドロップダウンメニューから、[Kubernetes] を選択します。

サポートされている Kubernetes または OpenShift プラットフォームを表示するには、[サポートされているプラットフォームの表示（Show Supported Platforms）] をクリックします。

ステップ 4 エージェントをインストールするテナントを選択します。

（注） Cisco Secure Workload SaaS クラスタでは、テナントを選択する必要はありません。

ステップ 5 Cisco Secure Workload との通信に HTTP プロキシが必要な場合は、[はい（Yes）] を選択し、有効なプロキシ URL を入力します。

ステップ 6 [ダウンロード（Download）] をクリックし、ファイルをローカルディスクに保存します。

ステップ 7 Kubernetes API サーバーにアクセスでき、デフォルトのコンテキスト、クラスタ、ユーザーとしての管理権限を持つ kubectl 構成ファイルが存在する Linux マシンで、インストーラスクリプトを実行します。

インストーラは、デフォルトの場所（~/.kube/config）からのファイルの読み取りを試みます。ただし、--kubeconfig コマンドラインオプションを使用して明示的に指定できます。

インストールスクリプトにより、インストールされた Cisco Secure Workload エージェントの Daemonset とポッドを確認するための説明が提供されます。



（注） ダウンロード前にエージェント インストーラ ページで構成された HTTP プロキシは、Cisco Secure Workload エージェントが Cisco Secure Workload クラスタに接続する方法のみを制御します。この設定は、Kubernetes または OpenShift ノードによる Docker イメージの取得方法には影響しません。これらのノードのコンテナランタイムでは、独自のプロキシ設定が使用されるためです。Docker イメージを Cisco Secure Workload クラスタからプルできない場合は、コンテナランタイムのイメージプルプロセスのデバッグが必要になり、適切な HTTP プロキシの追加が必要になる場合があります。

優れた可視性のための Solaris エージェントのインストール

Solaris エージェントのインストールの要件と前提条件

- 「サポートされているプラットフォームと要件」を参照してください。
- サービスをインストールして実行するには、ルート権限が必要です。
- エージェントおよびログファイルのストレージ要件：1 GB。
- ホストを監視しているセキュリティアプリケーションでセキュリティの除外を設定することにより、他のセキュリティアプリケーションがエージェントのインストールやエージェントのアクティビティをブロックしないようにします。「セキュリティの除外」を参照してください。

エージェントスクリプトインストーラ方式を使用したSolarisエージェントのインストール

インストールされた Solaris エージェントは、優れた可視性とプロセスやパッケージの可視性の両方をサポートします。

手順

ステップ 1 エージェントのインストール方式に移動するために、次の手順を実行します。

- 初めてのユーザーの場合は、クイックスタートウィザードを起動し、[エージェントのインストール（Install Agents）] をクリックします。
- 左側のペインで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックし、[インストーラ（Installer）] タブを選択します。

ステップ 2 [エージェントスクリプトインストーラ（Agent Script Installer）] をクリックします。

ステップ 3 [プラットフォームの選択（Select Platform）] ドロップダウンメニューから、[Solaris] を選択します。

サポートされている Solaris プラットフォームを表示するには、[サポートされているプラットフォームの表示（Show Supported Platforms）] をクリックします。

ステップ 4 エージェントをインストールするテナントを選択します。

（注） Cisco Secure Workload SaaS クラスタでは、テナントを選択する必要はありません。

ステップ 5 （オプション）ワークロードにラベルを割り当てるには、ラベルキーを選択し、ラベル値を入力します。

インストールされたエージェントがホストで IP アドレスを報告すると、このホストによって報告された IP に割り当てられているアップロード済みの別の CMDB ラベルとともに、ここで選択されたインストーラ CMDB ラベルが新しい IP アドレスに自動的に割り当てられます。アップロード済みの CMDB ラベルとインストーラ CMDB ラベルの間で競合が発生した場合は、次のようになります。

- 正確な IP アドレスに割り当てられたラベルは、サブネットに割り当てられたラベルよりも優先されます。
- 正確な IP アドレスに割り当てられた既存のラベルは、インストーラの CMDB ラベルよりも優先されます。

ステップ 6 Cisco Secure Workload との通信に HTTP プロキシが必要な場合は、[はい（Yes）] を選択し、有効なプロキシ URL を入力します。

ステップ 7 [インストーラの有効期限（Installer expiration）] セクションで、利用可能なオプションから 1 つを選択します。

- [有効期限なし（No expiration）]：インストーラスクリプトは何回も使用できます。

- 1 回のみ：インストーラスクリプトは 1 回のみ使用できます。
- 時間制限：インストーラスクリプトを使用できる日数を設定できます。
- 展開数：インストーラスクリプトを使用できる回数を設定できます。

ステップ 8 [ダウンロード（Download）] をクリックし、ファイルをローカルディスクに保存します。

ステップ 9 Solaris ホストでインストーラ シェル スクリプトをコピーし、`chmod u+x tetration_installer_default_sensor_solaris.sh` コマンドを実行してスクリプトに実行権限を付与します。

（注） スクリプト名は、選択したエージェントのタイプと範囲によって異なる場合があります。

ステップ 10 エージェントをインストールするために、ルート権限で `./tetration_installer_default_sensor_solaris.sh` コマンドを実行します。

（注） エージェントがテナントにすでにインストールされている場合は、インストールを続行できません。

スクリプトの使用方法の詳細で指定されているように、事前チェックを実行することを推奨します。

Solaris インストーラスクリプトの使用方法の詳細：

```
tetration_installer_default_sensor_solaris.sh [--pre-check] [--skip-pre-check=<option>]
 [--no-install] [--logfile=<filename>] [--proxy=<proxy_string>] [--no-proxy] [--help]
[--version] [--sensor-version=<version_info>] [--ls] [--file=<filename>]
[--save=<filename>] [--new] [--reinstall] [--unpriv-user] [--force-upgrade]
[--upgrade-local] [--upgrade-by-uuid=<filename>] [--basedir=<basedir>]
[--logbasedir=<logbdir>] [--tmpdir=<tmp_dir>] [--visibility] [--golden-image]
  --pre-check: run pre-check only
  --skip-pre-check=<option>: skip pre-installation check by given option; Valid options
 include 'all', 'ipv6' and 'enforcement'; e.g.: '--skip-pre-check=all' will skip all
pre-installation checks; All pre-checks will be performed by default
  --no-install: will not download and install sensor package onto the system
  --logfile=<filename>: write the log to the file specified by <filename>
  --proxy=<proxy_string>: set the value of CL_HTTPS_PROXY, the string should be formatted
 as http://<proxy>:<port>
  --no-proxy: bypass system wide proxy; this flag will be ignored if --proxy flag was
provided
  --help: print this usage
  --version: print current script's version
  --sensor-version=<version_info>: select sensor's version; e.g.:
'--sensor-version=3.4.1.0'; will download the latest version by default if this flag was
 not provided
  --ls: list all available sensor versions for your system (will not list pre-3.1
packages); will not download any package
  --file=<filename>: provide local zip file to install sensor instead of downloading it
 from cluster
  --save=<filename>: download and save zip file as <filename>
  --new: remove any previous installed sensor; previous sensor identity has to be removed
 from cluster in order for the new registration to succeed
  --reinstall: reinstall sensor and retain the same identity with cluster; this flag has
 higher priority than --new
```

```
  --unpriv-user=<username>: use <username> for unpriv processes instead of nobody
  --force-upgrade: force sensor upgrade to version given by --sensor-version flag; e.g.:
 '--sensor-version=3.4.1.0 --force-upgrade'; apply the latest version by default if
--sensor-version flag was not provided
  --upgrade-local: trigger local sensor upgrade to version given by --sensor-version
flag: e.g.: '--sensor-version=3.4.1.0 --upgrade-local'; apply the latest version by
default if --sensor-version flag was not provided
  --upgrade-by-uuid=<filename>: trigger sensor whose uuid is listed in <filename> upgrade
 to version given by --sensor-version flag; e.g.: '--sensor-version=3.4.1.0
--upgrade-by-uuid=/usr/local/tet/sensor_id'; apply the latest version by default if
--sensor-version flag was not provided
  --logbasedir=<log_base_dir>: instead of logging to /opt/cisco/secure-workload/log use
 <log_base_dir>. The full path will be <log_base_dir>/secure-workload
  --tmpdir=<tmp_dir>: instead of using /tmp use <tmp_dir> as temp directory
  --visibility: install deep visibility agent only; --reinstall would overwrite this
flag if previous installed agent type was enforcer
  --golden-image: install Cisco Secure Workload Agent but do not start the Cisco Secure
 Workload Services; use to install Cisco Secure Workload Agent on Golden Images in VDI
environment or Template VM. On VDI/VM instance created from golden image with different
 host name, Cisco Secure Workload Services will work normally
```

Solaris エージェントのインストールの確認

手順

ステップ 1 コマンド `sudo pkg list tet-sensor` を実行します。

ステップ 2 出力として単一のエントリがあることを確認します。これにより、Solarisエージェントがホストにインストールされていることが確認されます。
サンプル出力：

NAME (PUBLISHER)	VERSION	IFO
tet-sensor (cisco)	3.8.1.1	i--

（注） 指定された出力は、プラットフォームとアーキテクチャによって異なる場合があります。

（手動インストールのみ）ユーザー構成ファイルの更新

以下の手順は、次のすべてを含むインストールにのみ必要です。

- Secure Workload SaaS、または複数のテナントを持つオンプレミスクラスタ（デフォルトのテナントのみを使用するオンプレミスクラスタでは、この手順は不要）
- 手動インストール
- Linux または Windows プラットフォーム

エージェントを Secure Workload クラスタに登録するには、クラスタ アクティベーション キーが必要です。さらに、エージェントがクラスタに到達するために HTTPS プロキシが必要な場合は、プロキシを指定する必要があります。



（注） Windows環境では、手動インストール時にアクティベーションキーとプロキシオプションを使用する場合、user.cfg を手動で構成する必要はありません。

インストールの前に、ユーザー構成ファイルで必要な変数を設定します。

手順

ステップ 1 アクティベーションキーの取得：**[管理（Manage）]**>**[エージェント（Agents）]**に移動し、[インストーラ（Installer）] タブをクリックし、[従来のパッケージインストーラを使用した手動インストール（Manual Install using classic packaged installers）] をクリックしてから、[エージェントアクティベーションキー（Agent Activation Key）] をクリックします。

ステップ 2 Secure Workload エージェントのインストールフォルダにある `user.cfg` ファイルを開いて編集します（例：Linux の場合は `/usr/local/tet`、Windows の場合は `C:\\Program Files\\Cisco Tetration`）。このファイルには、各行に 1 つずつ「key=value」の形式で変数のリストが含まれています。

ステップ 3 アクティベーションキーを **ACTIVATION_KEY** 変数に追加します。例：
```
ACTIVATION_KEY=7752163c635ef62e6568e9e852d07bd21bfd60d0
```

ステップ 4 エージェントに HTTPS プロキシが必要な場合は、**HTTPS_PROXY** 変数を使用して **http** プロトコルプロキシサーバーとポートを追加します。例：
```
HTTPS_PROXY=http://proxy.my-company.com:80
```

その他のエージェント同様のツール

AnyConnect エージェント

Network Visibility Module (NVM) を備えた Cisco AnyConnect セキュア モビリティ エージェントでサポートされるプラットフォームです。追加の Secure Workload エージェントは必要ありません。AnyConnect コネクタは、これらのエージェントを登録し、フロー観測データ、インベントリデータ、およびラベルをSecure Workload にエクスポートします。詳細については、「AnyConnect コネクタ」を参照してください。

Windows、Mac、Linux プラットフォームについては、『Cisco AnyConnect セキュア モビリティ クライアント データ シート』[英語] を参照してください。

ISE エージェント

Cisco Identity Services Engine（ISE）に登録されたエンドポイント。このエンドポイントに Secure Workload エージェントは必要ありません。ISE コネクタは、ISE アプライアンスの pxGrid サー

ビスを介して ISE からエンドポイントに関するメタデータを収集します。エンドポイントを Secure Workload で ISE エージェントとして登録し、このエンドポイントにインベントリのラベルをプッシュします。詳細については、「ISE コネクタ」を参照してください。

SPAN エージェント

SPAN エージェントは ERSPAN コネクタと連携します。詳細については、「ERSPAN コネクタ」を参照してください。

サードパーティ製品および追加のシスコ製品との統合

- Secure Workload で設定された外部オーケストレータを使用した統合。

  外部オーケストレータ （163 ページ）を参照してください。

- Secure Workload で設定されたコネクタを使用した統合。

  「コネクタとは」を参照してください。

接続情報

一般に、エージェントがワークロードにインストールされると、Secure Workload クラスタでホストされているバックエンドサービスへの多数のネットワーク接続が開始されます。エージェントのタイプとその機能に応じて、接続数の表示は異なります。

次の表は、さまざまなエージェントタイプによって確立されたさまざまな固定接続を示しています。

表 *3:* エージェントの接続

エージェント タイプ	コンフィギュレーション サーバー	コレクタ	適用バックエンド
可視性（オンプレミス）	CFG-SERVER-IP:443	COLLECTOR-IP:5640	該当なし
可視性（SaaS）	CFG-SERVER-IP:443	COLLECTOR-IP:443	該当なし
enforcement （オンプレミス）	CFG-SERVER-IP:443	COLLECTOR-IP:5640	ENFORCER-IP:5660
適用（SaaS）	CFG-SERVER-IP:443	COLLECTOR-IP:443	ENFORCER-IP:443
Docker イメージ	CFG-SERVER-IP:443	該当なし	該当なし

説明：

- CFG-SERVER-IP は、コンフィギュレーション サーバーの IP アドレスを表します。

- COLLECTOR-IP は、コレクタの IP アドレスを表します。詳細可視性エージェントと適用エージェントは、利用可能なすべてのコレクタに接続します。
- ENFORCER-IP は、適用エンドポイントの IP アドレスを表します。適用エージェントは、使用可能なエンドポイントのうち 1 つのみに接続します。
- Kubernetes/Openshift エージェントの展開の場合、インストールスクリプトにエージェントソフトウェアが含まれていません。エージェントソフトウェアを含む Docker イメージは、各 Kubernetes/Openshift ノードによって Secure Workload クラスタからプルされます。これらの接続は、コンテナランタイムイメージ取得コンポーネントによって確立され、CFG-SERVER-IP:443 が接続先になります。

（注）
- Secure Workload エージェントは、常にクライアントとして機能し、クラスタ内でホストされているサービスへの接続を開始しますが、サーバーとして接続を開始することはありません。
- 上記の固定接続に加えて、アップグレードがサポートされている特定のエージェントタイプの場合、エージェントは定期的に HTTPS 要求（ポート 443）をクラスタセンサー VIP に対して実行し、使用可能なパッケージを照会します。
- エージェントは NAT サーバーの背後に配置できます。

ワークロードがファイアウォールの背後にある場合や、ホスト ファイアウォール サービスが有効になっている場合は、クラスタへの接続が拒否される可能性があることに注意することが重要です。管理者は、適切なファイアウォールポリシーを作成して、そのような接続を許可する必要があります。

セキュリティの除外

ソフトウェアエージェントは、通常の動作中にホストのオペレーティングシステムと継続的に対話します。これにより、ウイルス対策やセキュリティエージェントなどの、ホストにインストールされている他のセキュリティ アプリケーションがアラームを発生させたり、Cisco Secure Workload エージェントのアクションをブロックしたりする可能性があります。そのため、エージェントが正常にインストールされ、機能していることを確認するには、ホストをモニターしているセキュリティ アプリケーションで必要なセキュリティの除外を構成する必要があります。

表 *4:* エージェントに関するセキュリティ除外のディレクトリ

ホスト OS	ディレクトリ
AIX	/opt/cisco/tetration
Linux	/usr/local/tet または /opt/cisco/tetration or <user chosen inst dir>

ホスト OS	ディレクトリ
Windows	C:\Program Files\Cisco Tetration

表 *5:* エージェントに関するセキュリティ除外のディレクトリ

ホスト OS	ディレクトリ
AIX	/opt/cisco/tetration
Linux	/usr/local/tet または /opt/cisco/tetration or <user chosen inst dir>
	/var/opt/cisco/secure-workload
Windows	C:\Program Files\Cisco Tetration
	C:\ProgramData\Cisco Tetration
[Solaris]	/opt/cisco/secure-workload

表 *6:* エージェントに関するセキュリティ除外のプロセス

ホスト OS	プロセス（Processes）
AIX	tet-engine、tet-sensor、tet-enforcer
Linux	tet-engine、tet-sensor、tet-enforcer、tet-main、enforcer
Windows	TetSenEngine.exe、TetSen.exe、TetEnfEgine.exe、TetEnfC.exe、TetEnf.exe、TetUpdate.exe、tet-main.exe

表 *7:* エージェントに関するセキュリティ除外のプロセス

ホスト OS	プロセス（Processes）
AIX	csw-agent
	tet-sensor
	tet-enforcer
	tet-main

ホスト OS	プロセス（Processes）
Linux	csw-agent
	tet-sensor
	tet-enforcer
	tet-main
	エンフォーサ
Windows	TetSenEngine.exe
	TetSen.exe
	TetEnfEgine.exe
	TetEnfC.exe
	TetEnf.exe
	TetUpdate.exe
	tet-main.exe
[Solaris]	csw-agent
	tet-sensor
	tet-main

表 *8 :* エージェントに関するセキュリティ除外のアクション

ホスト OS	アクション（Actions）
AIX	/dev/bpf*、/dev/ipl、/dev/kmem へのアクセス、curl の呼び出し
Linux	/proc のスキャン、netlink sockets を開く、curl、rpm/dpkg、ip[6]tables-save、ip[6]tables-restore、ipset-restore を呼び出す
Windows	レジストリへのアクセス、ファイアウォールイベントへの登録

表 *9:* エージェントに関するセキュリティ除外のアクション

ホスト **OS**	アクション（**Actions**）
AIX	/dev/bpf*、/dev/ipl、/dev/kmem へのアクセス
	cfg_ipf、curl、ipf、ippool、ipfstat lslpp、lsfilt、prtconf の呼び出し
	/proc のスキャン
Linux	curl、ip[6]tables-save、ip[6]tables-restore、rpm/dpkg の呼び出し
	/proc のスキャン、netlink ソケットを開く
Windows	レジストリへのアクセス
	ファイアウォールイベントへの登録
	c:\windows\system32\netsh.exe の呼び出し
[Solaris]	curl、lspp、pkg、smbios の呼び出し
	/proc のスキャン

表 *10:* エージェントに関するセキュリティ除外のスクリプトまたはバイナリ実行

ホスト **OS**	呼び出されるスクリプト/バイナリ
AIX	ksh: fetch_sensor_id.sh、check_conf_update.sh
Linux	bash：fetch_sensor_id.sh、check_conf_update.sh
Windows	cmd：fetch_sensor_id.cmd、check_conf_update.cmd、dmidecode.exe、npcap-installer.exe、sensortools.exe、signtool.exe

表 *11:* エージェントに関するセキュリティ除外のスクリプトまたはバイナリ実行

ホスト **OS**	呼び出されるスクリプト/バイナリ
AIX	-
Linux	-

ホスト OS	呼び出されるスクリプト/バイナリ
Windows	dmidecode.exe
	npcap-installer.exe
	sensortools.exe
	signtool.exe
[Solaris]	-

エージェントのサービスの管理

ソフトウェアエージェントは、サポートされているすべてのプラットフォームでサービスとして展開されます。このセクションでは、さまざまな機能とプラットフォームのサービスを管理する方法について説明します。



（注） 指定されていない限り、この項のすべてのコマンドは、実行するためにルート権限（Linux/Unix）または管理者権限（Windows）が必要です。

RHEL/CentOS/OracleLinux-6.x および Ubuntu-14 のサービス管理

次のことを行うには、次のコマンドを実行します。

- **サービスの開始**：`start csw-agent`
- **サービスの停止**：`stop csw-agent`
- **サービスの再起動**：`restart csw-agent`
- **サービスステータスの確認**：`status csw-agent`

RHEL/CentOS/OracleLinux-7.x 以降のサービスの管理

このコマンドは、以下でも適用できます。

- AlmaLinux、Rocky Linux- 8.x 以降
- Amazon Linux 2 以降
- Debian 8 以降
- SLES-12SPx 以降
- Ubuntu-16.04 以降

次のことを行うには、次のコマンドを実行します。

- **サービスの開始**：`systemctl start csw-agent`
- **サービスの停止**：`systemctl stop csw-agent`
- **サービスの再起動**：`systemctl restart csw-agent`
- **サービスステータスの確認**：`systemctl status csw-agent`

Windows サーバーまたは Windows VDI のサービス管理

次のことを行うには、次のコマンドを実行します。

- **サービスの開始**：`net start <service-name>`

  例：優れた可視性サービスの場合は **net start tetsensor**、適用サービスの場合は **net start tetenforcer**

- **サービスの停止**：`net stop <service-name>`

  例：優れた可視性サービスの場合は **net stop tetsensor**、適用サービスの場合は **net stop tetenforcer**

- **サービスの再起動**：

  1. `net stop <service-name>`
  2. `net start <service-name>`

- **サービスステータスの確認**：`sc query <service-name>`

  例：優れた可視性サービスの場合は **sc query tetsensor**、適用サービスの場合は **sc query tetenforcer**

AIX のサービスの管理

次のことを行うには、次のコマンドを実行します。

- **サービスの開始**：`startsrc -s csw-agent`
- **サービスの停止**：`stopsrc -s csw-agent`
- **サービスの再起動**：

  1. `stopsrc -s csw-agent`
  2. `startsrc -s csw-agent`

- **サービスステータスの確認**：`lssrc -s csw-agent`

Kubernetes Agent インストールでのサービス管理

- **サービスの開始/停止**：エージェントは個別のサービスとしてではなく、クラスタ全体のデーモンセットとしてインストールされるため、特定のノードでエージェントを停止または開始することはできません。
- **ノード上のエージェントの再起動**：ノードで Cisco Secure Workload エージェントのポッドを見つけ、適切なKubernetesコマンドを実行して強制終了します。ポッドは自動的に再起動します。
- **ポッドのステータスの確認**：`kubectl get pod -n tetration` または `oc get pod -n tetration`（OpenShift の場合）を実行すると、Kubernetes クラスタ内にあるすべての Cisco Secure Workload エージェントポッドのステータスが一覧で表示されます。

Solaris のサービスの管理

次のことを行うには、次のコマンドを実行します。

- **サービスの開始**：`svcadm enable csw-agent`
- **サービスの停止**：`svcadm disable csw-agent`
- **サービスの再起動**：`svcadm restart csw-agent`
- **サービスステータスの確認**：`svcs -l csw-agent`

エージェントによるポリシーの適用

デフォルトでは、ワークロードにインストールされたエージェントにはポリシーを適用する機能がありますが、適用は無効になっています。準備ができたら、設定目的に基づいて、これらのエージェントが選択したホストにポリシーを適用するようにできます。

エージェントがポリシーを適用すると、ファイアウォールが、送信元、宛先、ポート、プロトコル、方向などのパラメータに基づいて特定のネットワークトラフィックを許可するかドロップするかを指定する、順序付けられた一連のルールが適用されます。ポリシーの詳細については、「ポリシー」を参照してください。

エージェントを使用した適用について：

- エージェントは、セキュアな TCP/SSL チャネルを介してポリシーを受け取ります。
- エージェントは特権ドメインで実行されます。エージェントは、Linux マシンでは root として実行され、Windows マシンでは SYSTEM として実行されます。
- プラットフォームによっては、ポリシーの適用が有効になっている場合、エージェントはファイアウォールを完全に制御したり、既存の設定済みルールと連携して動作したりできます。

- 適用オプションの詳細、およびエージェントを有効にしてポリシーを適用するように設定する方法については、エージェント設定プロファイルの作成 （90 ページ）を参照してください。

高度な詳細：

適用を有効にすると、エージェントがコントローラに接続できるようにするゴールデンルールがプログラムされます。エージェントは、TLS/SSL プロトコルを介した双方向の安全なチャネルを介してコントローラの適用フロントエンド（EFE）と通信します。コントローラからのメッセージは、ポリシーの生成者によって署名され、エージェントによって検証されます。

エージェントは、コントローラからプラットフォームに依存しないスキーマでポリシーを受信します。エージェントは、これらのプラットフォームに依存しないポリシーをプラットフォーム固有のポリシーに変換し、エンドポイントのファイアウォールをプログラムします。

また、ファイアウォールの状態をアクティブにモニターし、適用されたポリシーの逸脱を検出すると、キャッシュされたポリシーをファイアウォールに再度適用します。エージェントは、CPU やメモリなどのシステムリソースの消費もモニターします。

エージェントは、EFE 経由で定期的にステータスと統計レポートをコントローラに送信します。ステータスレポートには、プログラムされた最新のポリシーステータス（成功、失敗、ある場合はエラー）が含まれます。統計レポートには、プラットフォームに応じたポリシー統計（許可またはドロップされたパケットおよびバイト数）が含まれます。

Linux プラットフォームでのエージェントの適用

Linux プラットフォームでは、エージェントは iptables/ip6tables/ipset を使用してネットワークポリシーを適用します。エンドホストでエージェントを有効にすると、デフォルトで iptables を制御およびプログラムします。IPv6 ネットワークスタックが有効になっている場合は、ip6tables を介して IPv6 ファイアウォールを制御します。

Linux iptables/ip6tables

Linux カーネルには、IPv4 および IPv6 パケットフィルタルールのテーブルをセットアップ、維持、および検査するために使用される iptables および ip6tables があります。これは、さまざまな事前定義されたテーブルで構成されています。各テーブルには、事前定義されたチェーンが含まれており、ユーザー定義のチェーンを含めることもできます。これらのチェーンには一連のルールが含まれており、これらの各ルールでパケットの一致基準を指定します。事前定義されたテーブルには、raw、mangle、filter、nat が含まれます。事前定義されたチェーンには、INPUT、OUTPUT、FORWARD、PREROUTING、および POSTROUTING が含まれます。

Secure Workload エージェントは、パケットを許可またはドロップするルールを含むフィルタテーブルをプログラムします。フィルタテーブルは、事前定義されたチェーンである INPUT、OUTPUT、および FORWARD で構成されます。これらに加えて、エージェントはカスタム TA チェーンを追加して、コントローラからポリシーを分類および管理します。これらの TA チェーンには、ポリシーから派生した Secure Workload ルールと、エージェントによって生成されたルールが含まれています。エージェントは、プラットフォームに依存しないルールを受信すると、それらを解析して iptable/ip6table/ipset ルールに変換し、これらのルールをフィルタテーブ

ルの TA 定義チェーンに挿入します。ファイアウォールのプログラミング後、エージェントはファイアウォールを監視し、ルールやポリシーの逸脱がないか確認します。逸脱がある場合は、ファイアウォールを再プログラミングします。また、ファイアウォールでプログラムされたポリシーを追跡し、ポリシーのステータスを定期的にコントローラに報告します。

この動作を表す例を次に示します。

プラットフォームに依存しないネットワーク ポリシー メッセージの一般的なポリシーの構成は次のとおりです。

```
source set id: “test-set-1”
destination set id: “test-set-2”
source ports: 20-30
destination ports: 40-50
ip protocol: TCP
action: ALLOW
. . .
set_id: “test-set-1”
     ip_addr: 1.2.0.0
     prefix_length: 16
     address_family: IPv4
set_id: “test-set-2”
     ip_addr: 3.4.0.0
     prefix_length: 16
     address_family: IPv4
```

エージェントは、他の情報とともに、このポリシーを処理し、プラットフォーム固有の ipset および iptabel ルールに変換します。

```
ipset rule:
Name: ta_f7b05c30ffa338fc063081060bf3
Type: hash:net
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 16784
References: 1
Members:
1.2.0.0/16
Name: ta_1b97bc50b3374829e11a3e020859
Type: hash:net
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 16784
References: 1
Members:
3.4.0.0/16
iptables rule:
TA_INPUT -p tcp -m set --match-set ta_f7b05c30ffa338fc063081060bf3 src -m set --match-
‹→set ta_1b97bc50b3374829e11a3e020859 dst -m multiport --sports 20:30 -m multiport --
‹→dports 40:50 -j ACCEPT
```

警告

ipset カーネルモジュール

エージェント構成プロファイルで [適用（Enforcement）] が有効で、[ルールの維持（Preserve Rules）] が無効になっている場合、Linux ホストで実行されている対象のエージェントにより、ipset カーネルモジュールの max_sets 設定が十分大きいことが確認されます。変更が必要な場合、エージェントは ipset カーネルモジュールを新しい max_sets 値でリロードします。代わりに [ルールの維持（Preserve Rules）] が有効になっている場合、エージェントにより現在の ipset

モジュールの max_sets 値がチェックされますが、変更はされません。現在設定されている max_sets 値は、cat /sys/module/ip_set/parameters/max_sets で確認できます。

ホストファイアウォールのバックアップ

エージェント構成プロファイルで初めて [適用（Enforcement）] が有効になると、Linux ホストで実行されている対象のエージェントは、ホストのファイアウォールを制御する前に、ipset および ip[6] テーブルの現在のコンテンツを /opt/cisco/tetration/backup に保存します。

適用の設定を続けて無効化または有効化しても、新しいバックアップは生成されません。このディレクトリは、エージェントのアンインストール時に削除されません。

WAF モードの Windows プラットフォームでのエージェントの適用

Windows プラットフォームでは、Secure Workload エージェントは Windows ファイアウォールを使用してネットワークポリシーを適用します。

セキュリティが強化された Windows ファイアウォール

次のようなタイプの設定に基づいてネットワークトラフィックを規制する、Windows のネイティブコンポーネントです。

- インバウンド ネットワーク トラフィックを規制するファイアウォールルール
- アウトバウンド ネットワーク トラフィックを規制するファイアウォール ルール
- ネットワークトラフィックの送信元と宛先の認証ステータスに基づくファイアウォール オーバーライド ルール
- IPSec トラフィックと Windows サービスに適用されるルール。

Secure Workload ネットワークポリシーは、インバウンドおよびアウトバウンドのファイアウォールルールを使用してプログラムされます。

Secure Workload ルールと Windows ファイアウォール

Windows プラットフォームでは、Secure Workload ネットワークポリシーは次のように適用されます。

1. プラットフォームに依存しないファイアウォールルールを、Secure Workload ネットワークポリシーから Windows ファイアウォールルールに変換します。
2. Windows ファイアウォールでルールをプログラムします。
3. Windows ファイアウォールがルールを適用します。
4. Windows ファイアウォールとそのルールセットの状態をモニターします。変更が検出された場合は、逸脱をレポートし、Windows ファイアウォールで Secure Workload ネットワークポリシーをリセットします。

セキュリティ プロファイル（Security Profiles）

Windows ファイアウォールは、ホストが現在接続しているネットワークに基づいてルールをグループ化します。これらはプロファイルと呼ばれ、次のような 3 つのプロファイルがあります。

- ドメインプロファイル
- プライベートプロフィール
- パブリックプロファイル

Secure Workload ルールはすべてのプロファイルにプログラムされていますが、アクティブなプロファイル内のルールのみが継続的に監視されます。

効果的な設定および混合リストのポリシー

Windows ファイアウォールのルールセットは、優先順位に基づいて順序付けられていません。複数のルールが 1 つのパケットに一致する場合、それらのルールのうち最も制限の厳しいものが有効になります。つまり、拒否ルールが許可ルールより優先されます。詳細については、「Microsoft TechNet に関する記事」[英語] を参照してください。

エージェント適用セクションからの混合リスト（許可および拒否の両方）ポリシーの例について考えてみます。

```
1. ALLOW 1.2.3.30 tcp port 80
2. ALLOW 1.2.3.40 udp port 53
3. BLOCK 1.2.3.0/24 ip
4. ALLOW 1.2.0.0/16 ip
5. Catch-all: DROP ingress, ALLOW egress
```

ホスト 1.2.3.30 の TCP ポート 80 宛てのパケットがファイアウォールに到達すると、上記のすべてのルールに一致しますが、最も制限の厳しいルール 3 が適用され、パケットはドロップされます。この動作は、ルールが順番に評価されてルール 1 が適用され、パケットが許可されるという期待に反します。

この動作の違いは、前述の Windows ファイアウォールの設計により、Windows プラットフォームで予期されるものです。この動作は、異なるルールアクションが設定された重複するルールを持つ混合リストポリシーで観察できます。

次に例を示します。

```
1. ALLOW 1.2.3.30 tcp
2. BLOCK 1.2.3.0/24 tcp
```

他のファイアウォールやポリシーからの干渉

Secure Workload ネットワークポリシーを意図したとおりに適用するには、エージェントに Windows ファイアウォールの完全かつ排他的な制御を許可することを推奨します。次の場合、エージェントはポリシーを確実に適用できません。

- サードパーティのファイアウォールが存在する (Windows ファイアウォールは、ホスト上でアクティブなファイアウォール製品である必要があります）。

- ファイアウォールが現在のプロファイルに対して無効になっている。
- 競合するファイアウォール設定がグループポリシーを使用して展開されている。以下の設定で競合が発生する場合があります。
  - ファイアウォールルール
  - 現在のプロファイルに設定されたデフォルトのインバウンドまたはアウトバウンドアクションがポリシーのキャッチオールルールとは異なる場合
  - ファイアウォールが現在のプロファイルで無効になっている場合

ステートフル適用

Windows Advanced Firewall は**ステートフル**ファイアウォールと見なされます。つまり、特定のプロトコル（TCP など）に対して、ファイアウォールは内部状態の追跡を維持して、ファイアウォールに到達する新しいパケットが既知の接続に属しているかどうかを検出します。既知の接続に属するパケットは、ファイアウォールルールを調べることなく許可されます。これにより、着信テーブルと発信テーブルの両方でルールを確立することなく、双方向通信が可能になります。

たとえば、Web サーバーについて、**ポート 443 へのすべての TCP 接続を受け入れる** というルールを考えてみます。

意図は明確です。ポート 443 でサーバーへのすべての TCP 接続を受け入れ、サーバーがクライアントに向けて通信できるようにします。この場合、着信テーブルには、ポート 443 での TCP 接続を許可するルール 1 つのみを挿入します。発信テーブルに挿入する必要のあるルールはありません。これは Windows Advanced Firewall によって暗黙的に行われるためです。

状態追跡は、明示的な**接続**が確立および維持される一部のプロトコルにのみ適用されることに注意してください。他のプロトコルの場合、双方向通信を可能にするために、着信ルールと発信ルールの両方をプログラムする必要があります。

適用が有効な場合、プロトコルが TCP の場合、特定の具象ルールは**ステートフル**としてプログラムされます（エージェントは、コンテキストに基づいて、ルールを着信テーブルまたは発信テーブルのどちらに挿入するかを決定します）。他のプロトコル（**ANY** を含む）の場合、着信ルールと発信ルールの両方がプログラムされます。

警告

ホストファイアウォールのバックアップ

エージェント設定プロファイルで初めて適用が有効になると、Windows ホストで実行中の対象エージェントは、ホストのファイアウォール制御を開始する前に、現在の Windows 高度ファイアウォールのコンテンツを *Program- Data\Cisco\Tetration\backup* にエクスポートします。適用の設定を続けて無効化または有効化しても、新しいバックアップは生成されません。このディレクトリは、エージェントのアンインストール時に削除されません。

WFP モードでの Windows プラットフォームにエージェントを適用

Windows プラットフォームでは、エージェントは WFP フィルタをプログラミングするネットワークポリシーを適用します。Windows の高度なファイアウォールは、ネットワークポリシーの設定には使用されません。

WFP（Windows フィルタリング プラットフォーム）

WFP（Windows Filtering Platform）は、ネットワークトラフィック処理フィルタを設定するために Microsoft が提供する一連の API です。ネットワークトラフィック処理フィルタは、カーネルレベルの API およびユーザーレベルの API を使用して設定できます。WFP フィルタは、ネットワークレイヤ、トランスポートレイヤ、アプリケーションレイヤの適用（ALE）など、さまざまなレイヤで設定できます。Secure Workload WFP フィルタは、Windows ファイアウォールルールと同様に、ALE レイヤで設定されます。それぞれのレイヤには、重みの高いものから順に並べられた、いくつかのサブレイヤがあります。各サブレイヤ内で、フィルタは重みの高いものから順に並べられます。ネットワークパケットは、すべてのサブレイヤを通過します。各サブレイヤで、ネットワークパケットは、重みに基づいて一致するフィルタを通過し、許可またはブロックのアクションを返します。すべてのサブレイヤを通過した後、アクションに基づいてパケットが処理されます。ブロックアクションは許可をオーバーライドします。

WAF における WFP の利点

- Windows ファイアウォール設定の依存関係を回避
- GPO 制限なし
- 移行とポリシー復元の容易さ
- ユーザーによるポリシー順序の制御を実現
- Windows ファイアウォールの厳格なブロック優先のポリシー順序を回避
- ポリシー更新時の CPU オーバーヘッドの削減
- 効率的な 1:1 ポリシールールフィルタの作成
- 高速なシングルステップ更新

エージェントによる WFP のサポート

適用が WFP を使用するように設定されている場合、Secure Workload フィルタは Windows ファイアウォールルールをオーバーライドします。

WFP モードで、エージェントはさまざまな WFP オブジェクトを設定します。

- プロバイダー：フィルタ管理に使用されます。パケットフィルタリングには影響しません。GUID と名前が指定されます。

- サブレイヤー：サブレイヤーには、名前、GUID、および重みが指定されます。Secure Workload サブレイヤーは、Windows Advanced Firewall サブレイヤーよりも大きな重み付けで設定されます。
- フィルタ：フィルタには、名前、GUID、ID、重み、レイヤー ID、サブレイヤーキー、アクション（PERMIT/BLOCK）、およびフィルタ条件があります。WFP フィルタは、ゴールデンルール、セルフルール、ポリシールール用に設定されています。エージェントは、ポートスキャン防止フィルタも設定します。Secure Workload フィルタは、フラグ FWPM_FILTER_FLAG_CLEAR_ACTION_RIGHT を使用して設定されます。このフラグのため、Secure Workload フィルタのアクションを Microsoft ファイアウォールルールによって上書きすることはできません。Secure Workload ネットワークポリシールールごとに、方向（インバウンド/アウトバウンド）とプロトコルに基づいて、1 つ以上の WFP フィルタが設定されます。

TCP インバウンドポリシーの場合：

```
id: 14 , TCP Allow 10.195.210.184 Dir=In localport=3389
```

設定された WFP フィルタ

```
Filter Name:                    Secure Workload Rule 14
---------------------------------------------------
EffectiveWeight:                          18446744073709551589
LayerKey:                                 FWPM_LAYER_ALE_AUTH_LISTEN_V4
Action:                                   Permit
Local Port:                               3389
Filter Name:                              Secure Workload Rule 14
---------------------------------------------------
EffectiveWeight:                          18446744073709551589
LayerKey:                                 FWPM_LAYER_ALE_AUTH_RECV_ACCEPT_V4
Action:                                   Permit
RemoteIP:                                 10.195.210.184-10.195.210.184
```

Secure Workload エージェントは、インバウンド CATCH-ALL ポリシーの **Cisco Secure Workload デフォルトインバウンド**フィルタを設定します。Secure Workload エージェントは、アウトバウンド CATCH-ALL ポリシーの **Cisco Secure Workload デフォルトアウトバウンド**フィルタを設定します。

エージェント WFP のサポートと Windows ファイアウォール

- エージェントは、WAF ルールまたは WAF プロファイルを**監視しません**。
- エージェントはファイアウォールの状態を**監視しません**。
- エージェントでは、ファイアウォールの状態を有効にする**必要はありません**。
- エージェントは GPO ポリシーと**競合しません**。

効果的な設定および混合リストのポリシー

WFP モードでのエージェント適用は、混合リストまたはグレーリストのポリシーをサポートします。

エージェント適用セクションからの混合リスト（許可および拒否の両方）ポリシーの例を考えてみます。

```
1. ALLOW 1.2.3.30 tcp port 80-            wt1000
2. BLOCK 1.2.3.0/24 ip-                   wt998
3. ALLOW 1.2.0.0/16 ip-                   wt997
4. Catch-all: DROP ingress, ALLOWegress - wt996
```

ホスト 1.2.3.30 tcp ポート 80 に向かうパケットがファイアウォールに到達すると、ルール 1 に一致します。しかし、ホスト 1.2.3.10 に向かうパケットは、フィルタ 2 のためにブロックされます。ホスト 1.2.2.10 に向かうパケットは、フィルタ 3 によって許可されます。

ステートフル適用

Cisco Secure Workload の WFP フィルタは、ALE レイヤで構成されます。ネットワークトラフィックは、ソケットの connect()、listen()、および accept() 操作に対してフィルタリングされます。L4 接続に関連するネットワークパケットは、接続が確立されるとフィルタリングされなくなります。

構成された WFP フィルタの可視性

構成された Secure Workload WFP フィルタは、c:\program files\tetration\tetenf.exe を使用して表示できます。次のオプションがサポートされます。

- 「管理者」権限を使用して「cmd.exe」を実行します
- c:\program files\tetration\tetenf.exe -l -f <-verbose> <-output=outfile.txt> を実行する

または

- 「管理者」権限を使用して「cmd.exe」を実行します
- netsh wfp show filters を実行する
- 構成された Secure Workload フィルタについて filters.xml を確認する

WFP モードでステルスモードフィルタを無効にする

ステルスモードフィルタ（ポートスキャンフィルタ）を無効にする手順は次のとおりです。

手順

ステップ 1 \conf\enforcer.cfg を編集します

ステップ 2 disable_wfp_stealth_mode: 1 を追加します。

ステップ 3 ファイルを保存します。

ステップ 4 管理者権限で TetEnforcer サービスを再起動します。

a) コマンド `sc stop tetenforcer` を実行して、TetEnforcer サービスを停止します。

b) コマンド `sc start tetenforcer` を実行して、TetEnforcer サービスを開始します。

ステップ 5 次の手順で確認します。

a) 「管理者」権限を使用してコマンド `cmd.exe` を実行します。

b) コマンド `c:\program files\tetration\tetenf.exe -l -f <-verbose> <-output=outfile.txt>` を実行します。

```
“Tetration Internal Rule block portscan” filters are not configured.
```

設定済みの WFP フィルタの削除

設定された Secure Workload WFP フィルタは、c:\program files\tetration\tetenf.exe を使用して表示できます。フィルタを誤って削除しないようにするには、ユーザーが [トークン（token）] を指定する必要があります。削除コマンドを実行する際、yyyy は現在の年、mm は現在の月を数値形式で表します。たとえば、今日の日付が 2021 年 1 月 21 日の場合、トークンは [-token=202101] になります。

次のオプションがサポートされます。

- 「管理者」権限を使用して「cmd.exe」を実行します。
- 設定されたすべての Secure Workload フィルタを削除するには、c:\program files\tetration\tetenf.exe -d -f -all - token=<yyyymm> を実行します。
- 設定されたすべての Secure Workload WFP オブジェクトを削除するには、c:\program files\tetration\tetenf.exe -d -all -token=<yyyymm> を実行します。
- Secure Workload WFP フィルタを名前で削除するには、c:\program files\tetration\tetenf.exe -d -name=<WFP filter name> -token=<yyyymm> を実行します。

既知の制限事項

- 適用モードが WFP に設定されている場合、エージェント設定プロファイルの [ルールの保持（Preserve Rules）] 設定は無効になります。

Windows 属性のポリシーの設定

Windows ベースのワークロードにポリシーを適用する際の精度をさらに高めるために、次の方法でネットワークトラフィックをフィルタリングできます。

- アプリケーション
- サービス名（Service Name）
- ユーザー名（ユーザーグループありまたはなし）

これは、WAF および WFP モードでサポートされています。Windows OS ベースのフィルタは、生成されたネットワークポリシーでコンシューマフィルタとプロバイダーフィルタに分類されます。コンシューマフィルタは、コンシューマワークロードで開始されたネットワークトラ

フィックをフィルタリングします。プロバイダーフィルタは、プロバイダーワークロード宛てのネットワークトラフィックをフィルタリングします。

始める前に

この手順は、既存のポリシーを変更していることが前提となります。Windows OS ベースのフィルタを追加するポリシーを作成していない場合は、最初にそのポリシーを作成します。



重要 Windows 属性を含むポリシーについては、警告 （74 ページ）および既知の制限事項 （74 ページ）を参照してください。

手順

ステップ 1 [防御（Defend）] > [セグメンテーション（Segmentation）] の順に選択します。

ステップ 2 Windows OS ベースのフィルタを設定するポリシーを含む範囲をクリックします。

ステップ 3 ポリシーを編集するワークスペースをクリックします。

ステップ 4 [ポリシーの管理（Manage Policies）] をクリックします。

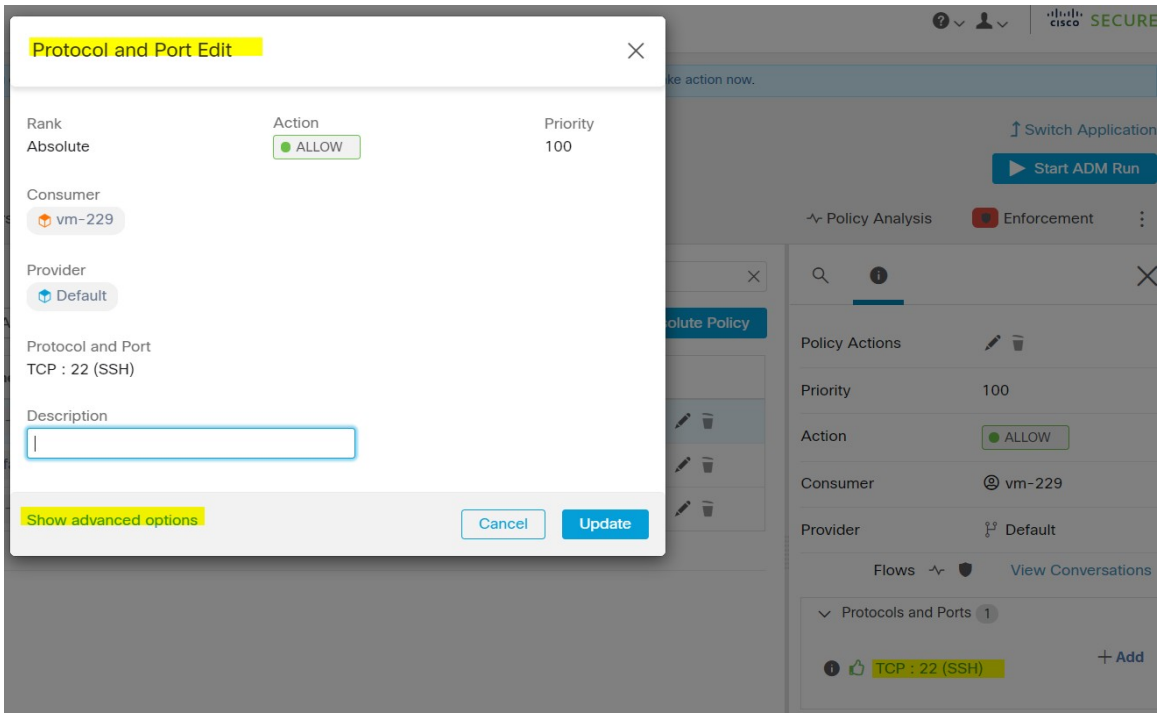
ステップ 5 編集するポリシーを選択します。

重要コンシューマとプロバイダーには、Windows ワークロードのみを含める必要があります。

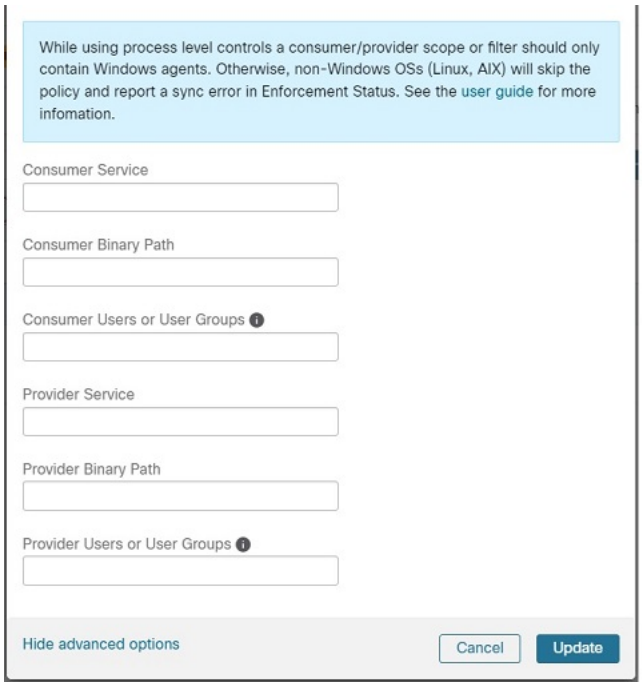
ステップ 6 編集するポリシーのテーブル行で、[プロトコルとポート（Protocols and Ports）] 列の既存の値をクリックします。

ステップ 7 右側のペインで、[プロトコルとポート（Protocols and Ports）] の下にある既存の値をクリックします。

この例では、[TCP : 22 (SSH)] をクリックします。



ステップ 8 [詳細オプションの表示（Show advanced options）] をクリックします。



ステップ 9 アプリケーション名、サービス名、またはユーザー名に基づいてコンシューマフィルタを設定します。

- アプリケーション名はフルパス名にする必要があります。

- サービス名は短いサービス名にする必要があります。
- ユーザー名は、ローカルユーザー名（tetter など）またはドメインユーザー名（sensor-dev@sensor-dev.com、sensor-dev\sensor-dev など）にできます。
- ユーザーグループは、ローカルユーザーグループ（管理者など）またはドメインユーザーグループ（domain users\\sensor-dev など）にできます。
- 複数のユーザー名および（または）ユーザーグループ名は「,」で区切って指定できます（例：sensor-dev\@sensor-dev.com,domain users\\sensor-dev）。
- サービス名とユーザー名は同時に設定できません。

ステップ 10 アプリケーション名、サービス名、またはユーザー名に基づいてプロバイダーフィルタを設定します。

前のステップのコンシューマフィルタと同じガイドラインに従います。

ステップ 11 必要に応じてバイナリへのパスを入力します。

たとえば、**c:\test\putty.exe** と入力します。

ステップ 12 [更新（Update）]をクリックします。

Windows OS ベースの推奨ポリシー設定

可能な場合は、常にポリシーでポートとプロトコルを指定します。任意のポート、任意のプロトコルを許可することはお勧めしません。

たとえば、ポートとプロトコルの制限を含めて生成されたポリシーは次のようになります。

```
 dst_ports {
  start_port: 22
  end_port: 22
  consumer_filters {
    application_name: "c:\\test\\putty.exe"
  }
 }}
 ip_protocol: TCP
```

対照的に、任意のプロトコルと任意のポートを使用して iperf.exe によって開始されたネットワーク接続を許可する場合、生成されるポリシーは次のようになります。

```
match_set {
  dst_ports {
    end_port: 65535
    consumer_filters {
      application_name: "c:\\test\\iperf.exe"
    }
  }
  address_family: IPv4
  inspection_point: EGRESS
```

```
  match_comment: "PolicyId=61008290755f027a92291b9d:61005f90497d4f47cedacb86:"
}
```

上記のフィルタの場合、Secure Workloadはプロバイダーのネットワークトラフィックを許可するポリシールールを次のように作成します。

```
match_set {
    dst_ports {
    end_port: 65535
  }
  address_family: IPv4
  inspection_point: INGRESS
  match_comment: "PolicyId=61008290755f027a92291b9d:61005f90497d4f47cedacb86:"
}
```

このネットワークルールは、プロバイダーのすべてのポートを開きます。任意のプロトコルを使用してOSベースのフィルタを作成しないことをお勧めします。

既知の制限事項

- Windows 2008 R2は、Windows OSベースのフィルタリングポリシーをサポートしていません。
- ネットワークポリシーは単一のユーザー名で設定できますが、MS Firewall UIは複数のユーザーをサポートします。

警告

- Windows OSベースのポリシーを使用している場合、コンシューマやプロバイダーの範囲またはフィルタにはWindowsエージェントのみを含める必要があります。そうしないと、Windows以外のOS（Linux、AIX）ではポリシーがスキップされ、適用ステータスで同期エラーが報告されます。
- フィルタリング基準が緩いWindows OSフィルタを作成しないでください。不要なネットワークポートが開く可能性があります。
- ネットワークフローのプロセスコンテキスト、ユーザーコンテキスト、またはサービスコンテキストに関する知識が乏しいか、知識がまったくないために、ポリシーにWindows OSベースのフィルタがある場合、ポリシー分析に矛盾が生じます。

Windows OSベースのフィルタリング属性を使用したポリシーの確認とトラブルシューティング

Windows OSベースのフィルタリング属性を使用する場合は、次のトピックを使用して、ポリシーが期待どおり動作することをワークロードで確認します。

Cisco TACは、必要に応じてこの情報を使用して、このようなポリシーのトラブルシューティングを行うことができます。

アプリケーション名に基づくポリシー

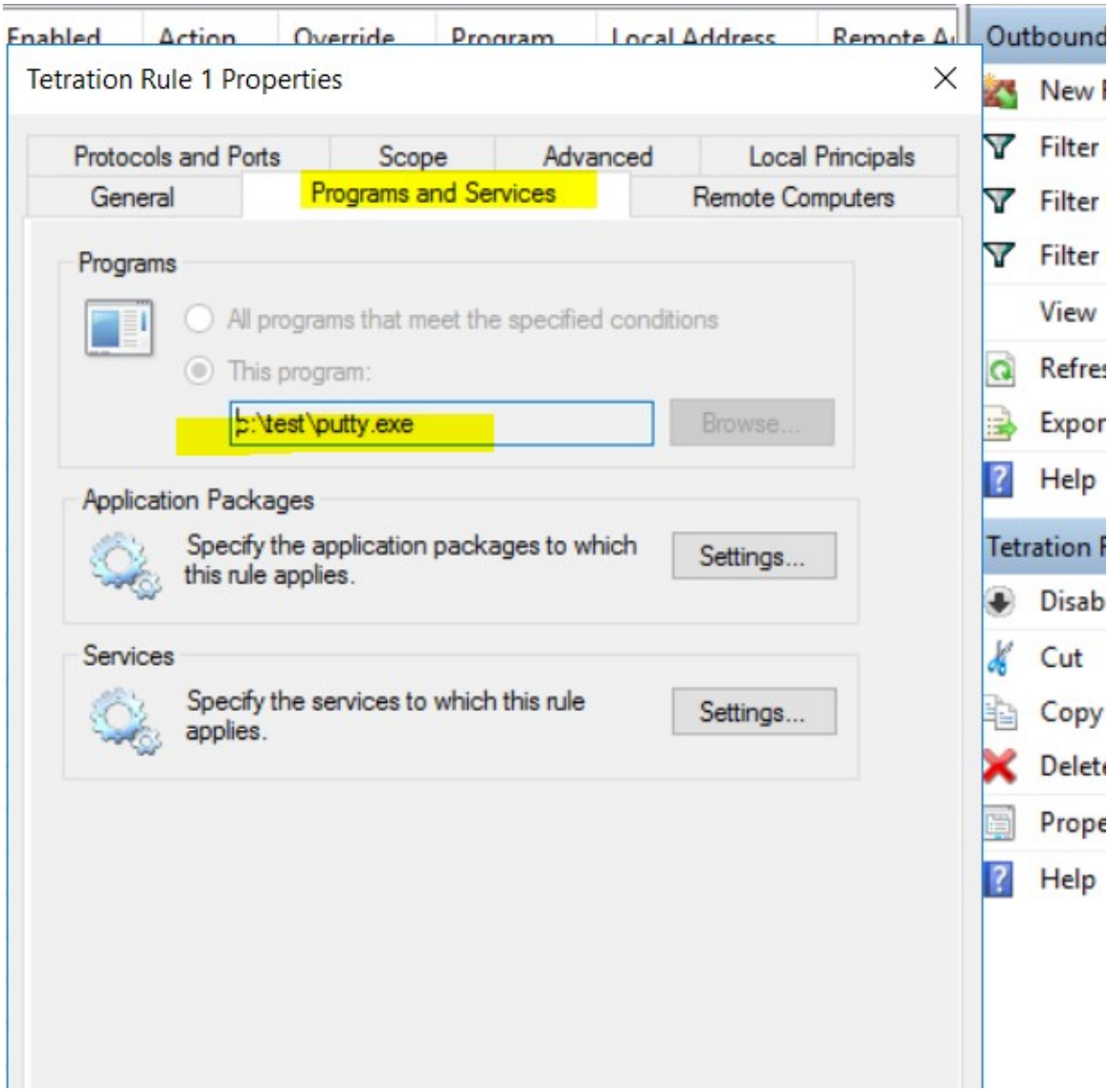
次の情報を使用して、アプリケーション名に基づく Windows OS ワークロードのポリシーを確認およびトラブルシューティングします。

次のセクションでは、**c:\test\putty.exe** として入力されたアプリケーションバイナリのワークロードにポリシーを表示させる方法について説明します。

アプリケーション名に基づくポリシーの例

```
dst_ports {
start_port: 22
end_port: 22
consumer_filters {
application_name: “c:\test\putty.exe”
}
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

生成されたファイアウォールルール



netsh を使用して生成されたフィルタ

高度なポリシーにフィルタが追加されていることをネイティブの Windows ツールで確認するには、次の手順を実行します。

- 「管理者」権限を使用して「cmd.exe」を実行します
- 「netsh wfp show filters」を実行します
- 出力ファイル filters.xml が、現在のディレクトリに生成されます。
- 出力ファイル（filters.xml）の FWPM_CONDITION_ALE_APP_ID でアプリケーション名を確認します。

```
<fieldKey>FWPM_CONDITION_ALE_APP_ID</fieldKey>
                      <matchType>FWP_MATCH_EQUAL</matchType>
                      <conditionValue>
                             <type>FWP_BYTE_BLOB_TYPE</type>
                             <byteBlob>
                                    <data>
‹→5c00640065007600690063006500 5c006800610072006400640069007300 6b0076006f006
‹→</data>
                                 <asString>\device\harddiskvolume2\temp\putty.exe</
‹→asString>
                 </byteBlob>
        </conditionValue>
```

tetenf.exe -l -f を使用して生成された WFP フィルタ

```
Filter Name:                    Secure Workload Rule 1
---------------------------------------------------------
EffectiveWeight:                18446744073709551592
LayerKey:                       FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                         Permit
RemoteIP:                       10.195.210.15-10.195.210.15
Remote Port:                    22
Protocol:                       6
AppID:                          \device\harddiskvolume2\test\putty.exe
```

アプリケーション名が無効な場合

- WAF モードでは、無効なアプリケーション名に対してファイアウォールルールが作成されます。
- WFP モードでは、無効なアプリケーション名に対して WFP フィルタは作成されませんが、NPC は拒否されません。エージェントは警告メッセージをログに記録し、残りのポリシールールを構成します。

サービス名に基づくポリシー

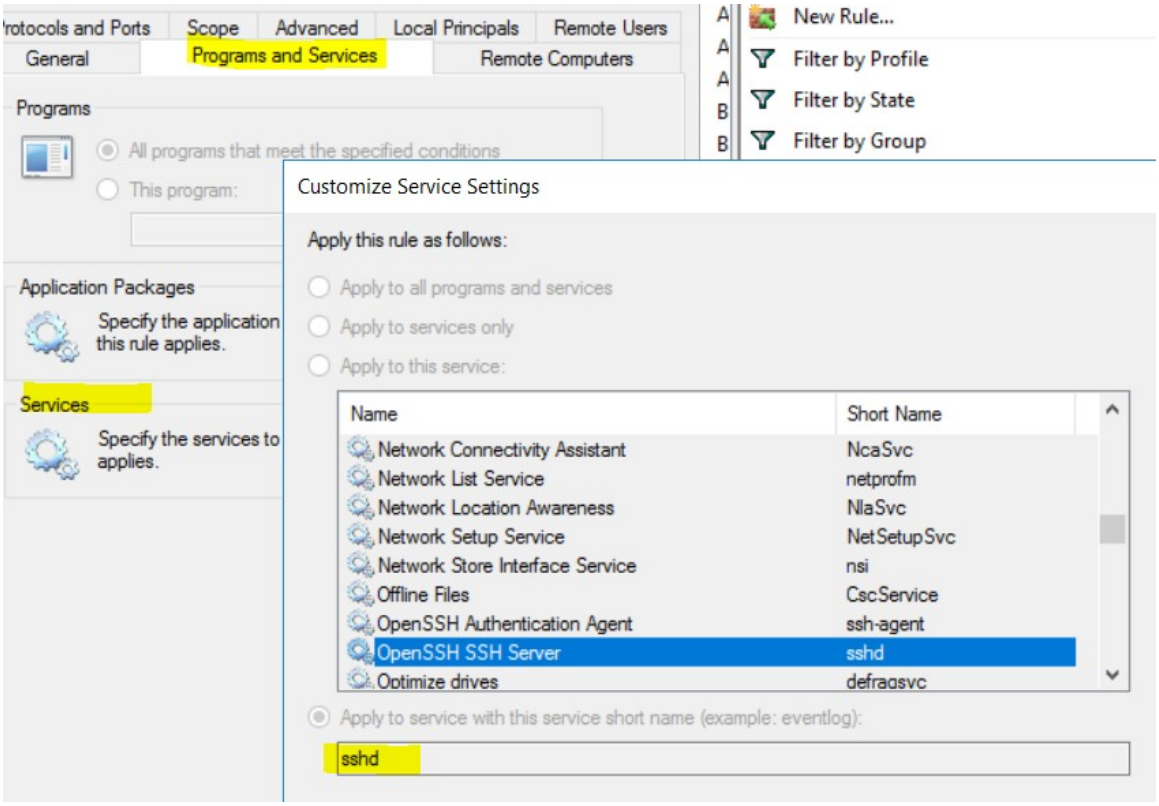
次の情報を使用して、サービス名に基づく Windows OS ワークロードのポリシーを確認およびトラブルシューティングします。

次のセクションでは、ワークロードにポリシーを表示させる方法について説明します。

サービス名に基づくサンプルポリシー

```
    dst_ports {
             start_port: 22
             end_port: 22
             provider_filters {
                    service_name: “sshd”
             }
         }}
         ip_protocol: TCP
         address_family: IPv4
         inspection_point: INGRESS
```

生成されたファイアウォールルール



netsh を使用して生成されたフィルタ

高度なポリシーにフィルタが追加されていることをネイティブの Windows ツールで確認するには、次の手順を実行します。

- 「管理者」権限を使用して「cmd.exe」を実行します
- 「netsh wfp show filters」を実行します
- 出力ファイル filters.xml が現在のディレクトリに生成されます
- 出力ファイル（filters.xml）でユーザー名の FWPM_CONDITION_ALE_USER_ID を確認します。

```
<item>
                    <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
                    <matchType>FWP_MATCH_EQUAL</matchType>
                    <conditionValue>
                              <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>

<sd>O:SYG:SYD:(A;;CCRC;;;S-1-5-80-3847866527-469524349-687026318-
→516638107)</sd>
                    </conditionValue>
</item>
```

tetenf.exe -l -f を使用して生成された WFP フィルタ

```
Filter Name:       Secure Workload Rule 3
-----------------------------------------------------
EffectiveWeight:            18446744073709551590
LayerKey:                   FWPM_LAYER_ALE_AUTH_RECV_ACCEPT_V4
Action:                     Permit
Local Port:                 22
Protocol:                   6
User or Service:            NT SERVICE\sshd
```

サービス名が不正な場合

- WAF モードで、存在しないサービス名に対してファイアウォールルールが作成されます
- WFP モードでは、存在しないサービス名に対して WFP フィルタは作成されません
- サービス SID タイプは「無制限」または「制限付き」である必要があります。サービスの種類が「なし」の場合、ファイアウォールルールと WFP フィルタを追加できますが、効果はありません。

  SID タイプを確認するには、次のコマンドを実行します。

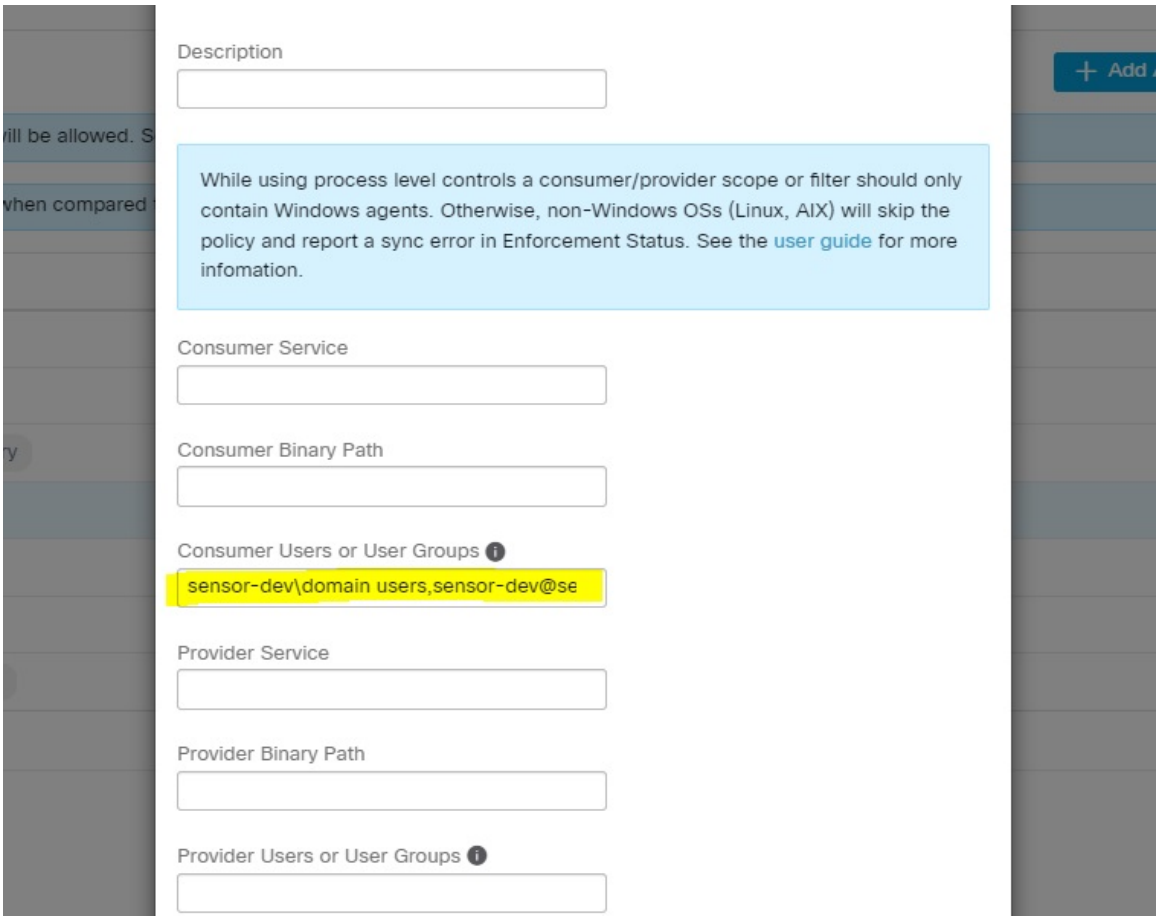
  ```
  sc qsidtype <service name>
  ```

ユーザーグループまたはユーザー名に基づくポリシー

次の情報を使用して、ユーザーグループ名の有無にかかわらず、ユーザー名に基づく Windows OS ワークロードのポリシーを確認およびトラブルシューティングします。

このトピックのセクションでは、ワークロードにポリシーを表示する方法について説明します。

このトピックの例は、次の情報を使用して設定されたポリシーに基づいています。



ユーザー名に基づくサンプルポリシー

```
dst_ports {
          start_port: 30000
          end_port: 30000
          provider_filters {
               user_name: “sensor-dev\sensor-dev”
           }
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

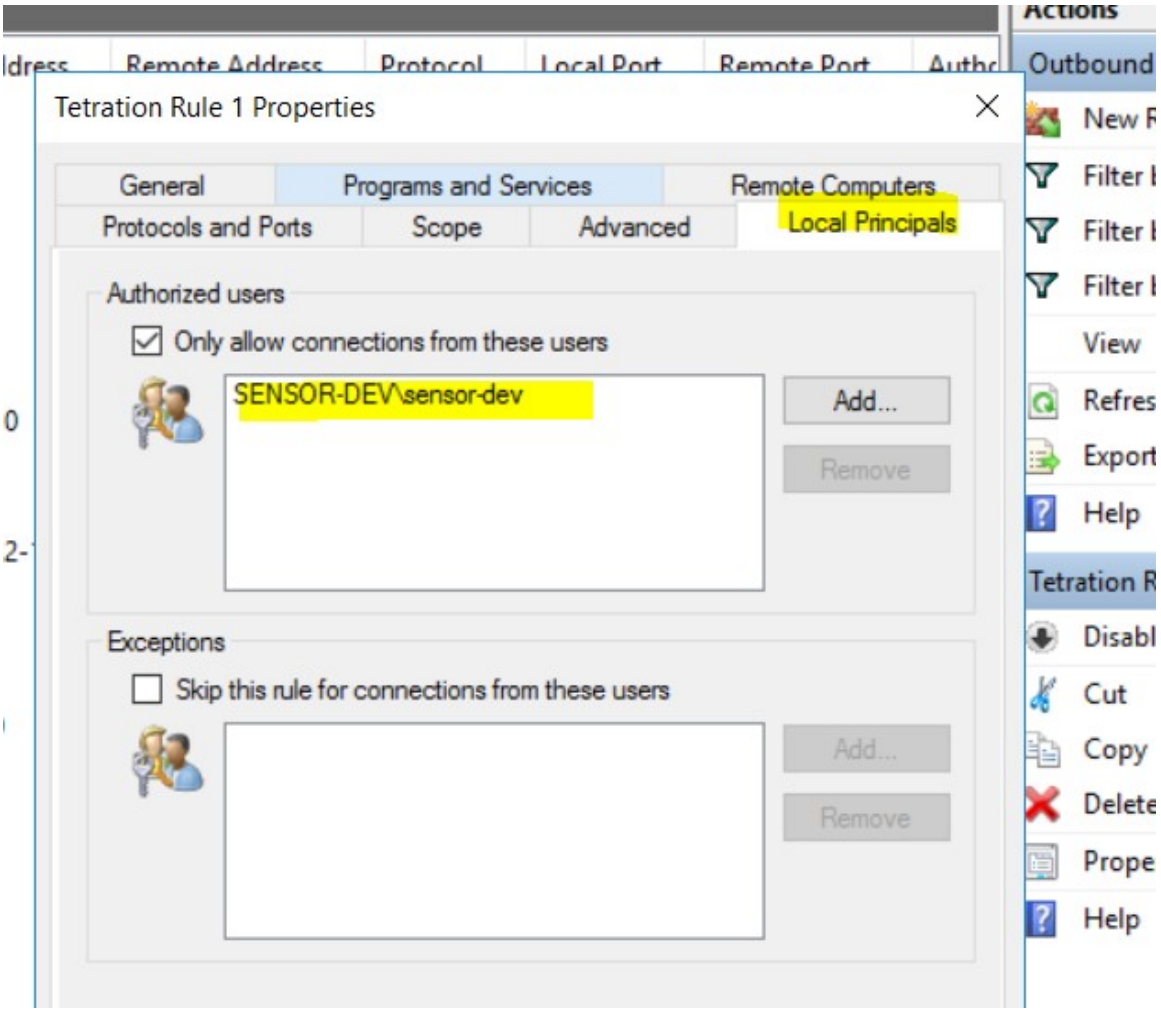
ユーザーグループとユーザー名に基づくサンプルポリシー

```
dst_ports {
start_port: 30000
end_port: 30000
provider_filters {
user_name: “sensor-dev\domain users,sensor-dev\sensor-dev”
}
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

生成されたファイアウォールルール

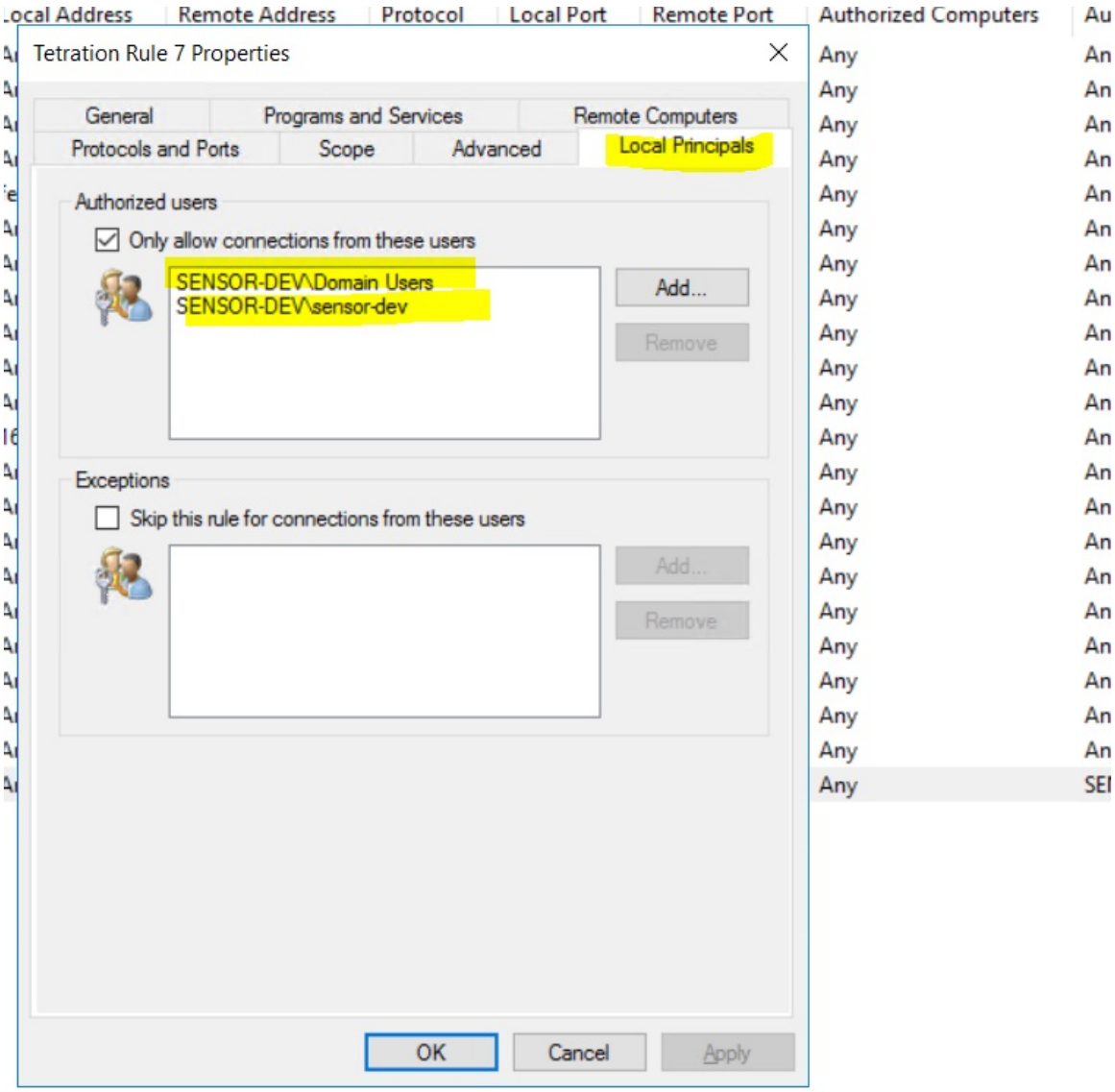
ユーザー名に基づくファイアウォールルール

例：ユーザー名 sensor-dev\\sensor-dev に基づくファイアウォールルール



ユーザーグループとユーザー名に基づくファイアウォールルール

例：ユーザー名 sensor-dev\\sensor-dev およびユーザーグループ domain users\\sensor-dev に基づくファイアウォールルール



netsh を使用して生成されたフィルタ

高度なポリシーにフィルタが追加されていることをネイティブの Windows ツールで確認するには、次の手順を実行します。

- 「管理者」権限を使用して「cmd.exe」を実行します
- 「netsh wfp show filters」を実行します
- 出力ファイル filters.xml が、現在のディレクトリに生成されます。
- 出力ファイル（filters.xml）でユーザー名の FWPM_CONDITION_ALE_USER_ID を確認します。

```
<item>
            <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
            <matchType>FWP_MATCH_EQUAL</matchType>
            <conditionValue>
                   <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
                  <sd>O:LSD:(A;;CC;;;S-1-5-21-4172447896-825920244-2358685150)</sd>

            </conditionValue>
</item>
```

tetenf.exe -l -f を使用して生成された WFP フィルタ

ユーザー名に基づくフィルタ

例：ユーザー名 SENSOR-DEV\sensor-dev に基づく WFP ルール

```
Filter Name:                    Secure Workload Rule 1
---------------------------------------------------
EffectiveWeight:                18446744073709551590
LayerKey:                       FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                         Permit
RemoteIP:                       10.195.210.15-10.195.210.15
Remote Port:                    30000
Protocol:                       6
User or Service:                SENSOR-DEV\sensor-dev
```

ユーザーグループとユーザー名に基づくフィルタ

例：ユーザー名 SENSOR-DEV\\sensor-dev およびユーザーグループ名 SENSOR-DEV\\Domain Users に基づく WFP ルール

```
Filter Name:          Secure Workload Rule 1
-----------------------------------------------------
EffectiveWeight:              18446744073709551590
LayerKey:                     FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                       Permit
RemoteIP:                     10.195.210.15-10.195.210.15
Remote Port:                  30000
Protocol:                     6
User or Service:              SENSOR-DEV\Domain Users, SENSOR-DEV\sensor-dev
```

ネットワークポリシールールにサービス名とユーザー名を設定することはできません。

ユーザー名またはユーザーグループが不正な場合

- ユーザー名またはユーザーグループが無効な場合、ネットワークポリシーはWindowsエージェントによって拒否されます。

AIX プラットフォームでのエージェントの適用

AIX プラットフォームでは、エージェントは IPFilter ユーティリティを使用してネットワークポリシーを適用します。エンドホストでエージェントを有効にすると、デフォルトでIPv4フィルタテーブル を制御およびプログラムします。IPv6 適用はサポートされていません。

IPFilter

AIX の IPFilter パッケージは、ファイアウォールサービスを提供するために使用されます。AIX では、カーネル拡張パックとして利用できます。また、カーネル拡張モジュール（/usr/lib/drivers/ipf）としてロードされます。このモジュールには、ipfilter ルールをプログラムするために使用される ipf、ippool、ipfstat、ipmon、ipfs、および ipnat ユーティリティが含まれており、各ルールでパケットの一致基準を指定します。詳細については、AIX の IPFilter man ページを参照してください。

適用が有効な場合、エージェントは IPFilter を使用して、IPv4 パケットを許可またはドロップするルールを含む IPv4 フィルタテーブルをプログラムします。エージェントは、それらのルールをグループ化して、コントローラからポリシーを分類および管理します。ルールには、ポリシーから派生した Secure Workload ルールと、エージェントによって生成されたルールが含まれます。

エージェントは、プラットフォームに依存しないルールを受け取ると、それらのルールを解析して ipfilter/ippool ルールに変換し、フィルタテーブルに挿入します。ファイアウォールのプログラミング後、適用エージェントはファイアウォールを監視し、ルールやポリシーの逸脱がないか確認します。逸脱がある場合は、ファイアウォールを再プログラミングします。また、ファイアウォールでプログラムされたポリシーを追跡し、ポリシーのステータスを定期的にコントローラに報告します。

プラットフォームに依存しないネットワーク ポリシー メッセージの一般的なポリシーの構成は次のとおりです。

```
source set id: "test-set-1"
destination set id: "test-set-2"
source ports: 20-30
destination ports: 40-50
ip protocol: UDP
action: ALLOW
...
set_id: "test-set-1"
   ip_addr: 1.2.0.0
   prefix_length: 16
   address_family: IPv4
set_id: "test-set-2"
   ip_addr: 5.6.0.0
   prefix_length: 16
   address_family: IPv4
```

エージェントは、他の情報とともに、このポリシーを処理し、プラットフォーム固有の ippool および ipfilter ルールに変換します。

```
    table role = ipf type = tree number = 51400
    { 1.2.0.0/16; };

    table role = ipf type = tree number = 75966
    { 5.6.0.0/16; };

    pass in quick proto udp from pool/51400 port 20:30 to pool/75966 port 40:50 group
TA_INPUT
```

警告

ホストファイアウォールのバックアップ

エージェント設定プロファイルで初めて適用が有効になると、AIXホストで実行されている対象のエージェントは、ホストのファイアウォールを制御する前に、ippoolとipfilterの現在のコンテンツを */opt/cisco/tetration/backup* に保存します。適用の設定を続けて無効化または有効化しても、新しいバックアップは生成されません。このディレクトリは、エージェントのアンインストール時に削除されません。

既知の制限事項

IPv6 適用はサポートされていません。

Allow ポリシーにより、既存の UDP 接続のトラフィックが中断される可能性があります。

エージェントのステータスと統計の確認

手順

ステップ 1 左側のメニューから、[管理（Manage）] > [エージェント（Agents）] の順に選択します。

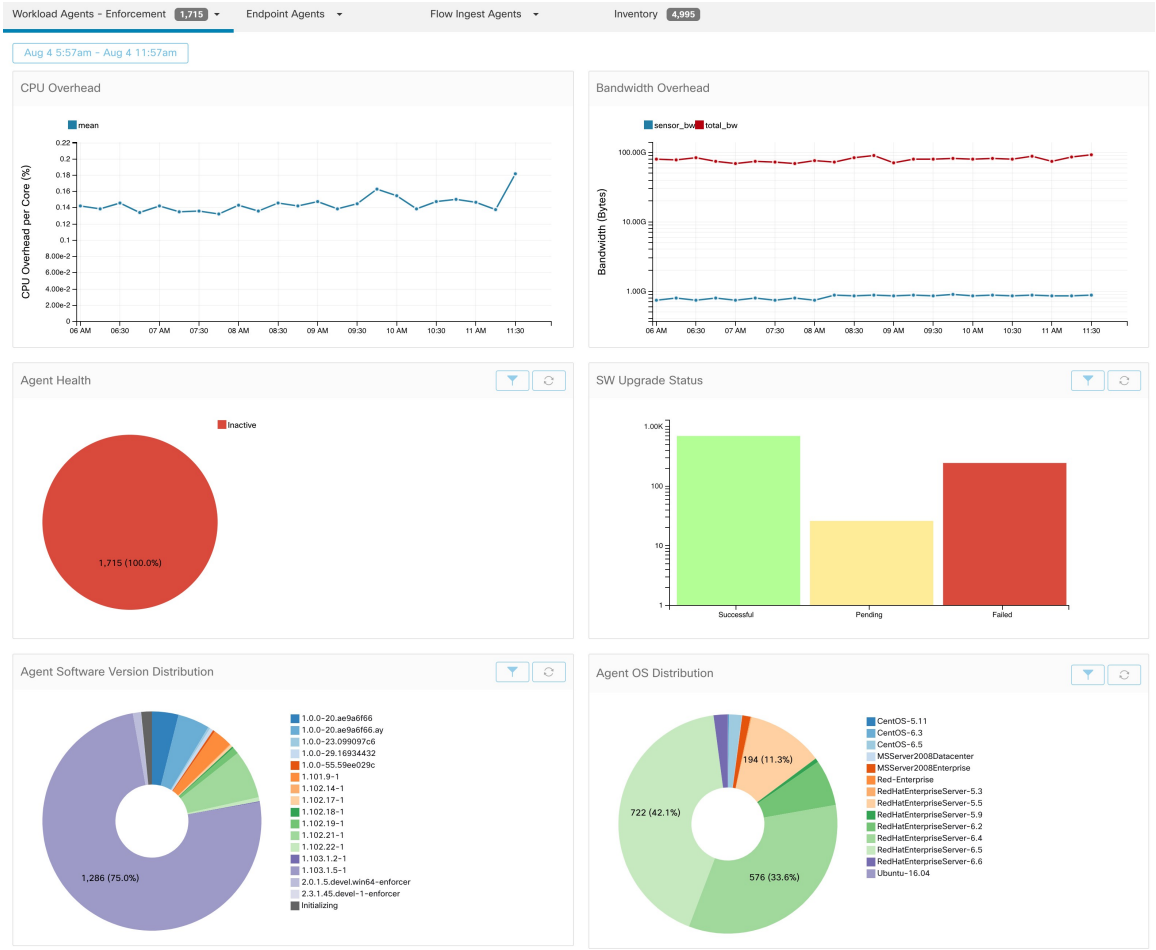
ステップ 2 [モニター（Monitor）] タブをクリックします。

ステップ 3 [分布（Distribution）] タブをクリックします。

ステップ 4 ページの上部からエージェントタイプをクリックします。

ステップ 5 このページでは、CPUオーバーヘッド、帯域幅オーバーヘッド、エージェントの状態、ソフトウェア アップデート ステータス、エージェント ソフトウェア バージョンの分布、エージェント OS の分布を確認できます。

図 *8 :* エージェントの分布ページ



ステップ 6 「適用ステータス」の項も参照してください。

エージェント詳細の表示

この手順では、1 つのワークロードとそこにインストールされたエージェントに関する多くの詳細が表示される、[ワークロードプロファイル（Workload Profile）] ページに移動する方法の 1 つを説明します。

手順

ステップ 1 [整理（Organize）] > [範囲とインベントリ（Scopes and Inventory）] を選択します。

ステップ 2 詳細を表示するワークロードを検索します。

ステップ 3 IP アドレスをクリックして、エージェントの正常性、IP アドレス、範囲、インベントリタイプ、適用グループ、実験グループ、ユーザーラベル、トラフィック量（合計バイト/合計パケット）などの豊富な詳細を表示します。

詳細については、ワークロードプロファイル（462 ページ）およびサブトピックを参照してください。

ソフトウェアエージェントの設定

ソフトウェアエージェント設定の要件と前提条件

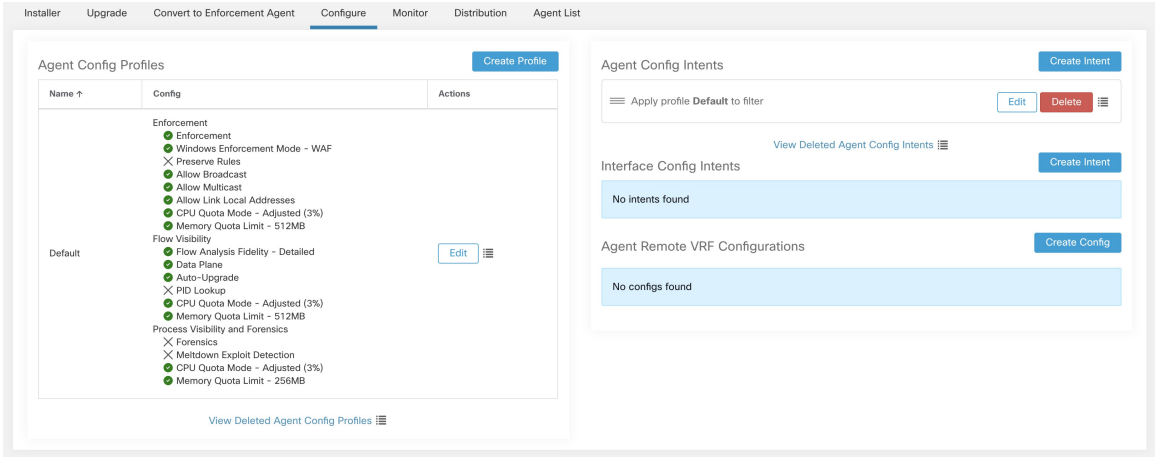
- 必要な Secure Workload ユーザーロールがあることを確認します。
  - サイト管理者
  - カスタマーサポート

  詳細は、ユーザーのロールとエージェント構成へのアクセス （87 ページ）を参照してください。
- お客様または別の許可されたユーザーが各ワークロードでエージェントサービスを実行するためのホストに対する権限を持っていることを確認します。「エージェントのサービスの管理」を参照してください。
- サポートされているプラットフォーム、要件、およびエージェントのインストール手順を確認します。「ソフトウェアエージェントの展開」を参照してください。

ユーザーのロールとエージェント構成へのアクセス

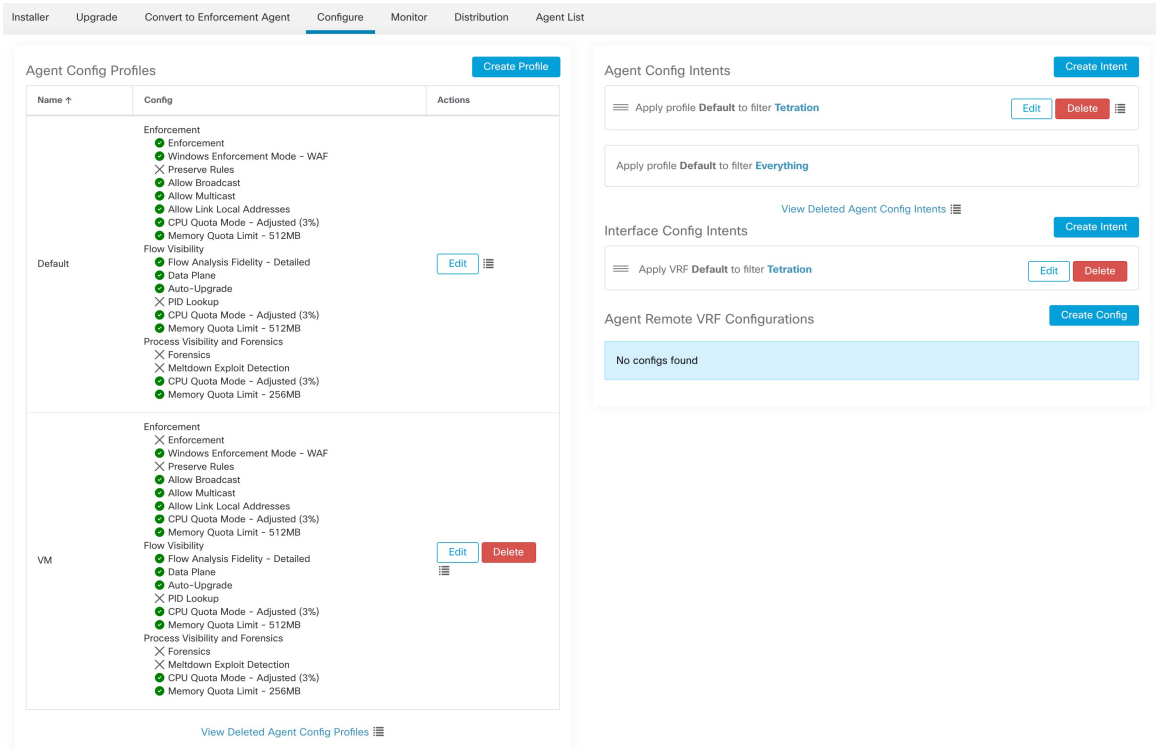
1. ルート範囲の所有者は、「構成プロファイル」の作成と「構成インテント」の仕様にのみアクセスできます。
2. ルート範囲の所有者は、所有する範囲にのみ関連付けられた構成プロファイルを作成し、所有するフィルタや範囲に該当するエージェントにのみ構成プロファイルを適用できます。

図 *9* : 範囲所有者ユーザーの *[*ソフトウェアエージェント構成（***Software Agent Config***）*]* タブ



3. サイト管理者ユーザーは、インターフェイス構成インテントとリモートVRF構成の指定を含む、[エージェント構成（Agent Config）]ページのすべてのコンポーネントにアクセスできます。

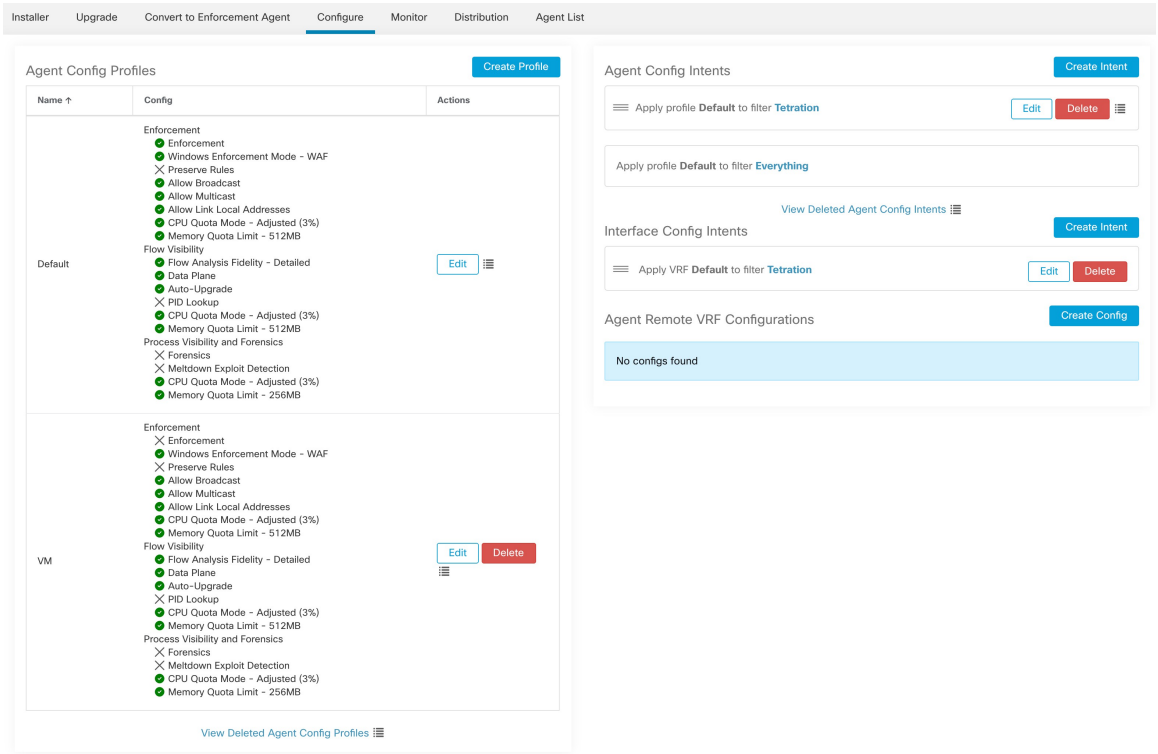
図 *10* : サイト管理者ユーザーの *[*ソフトウェアエージェント構成（***Software Agent Config***）*]* タブ

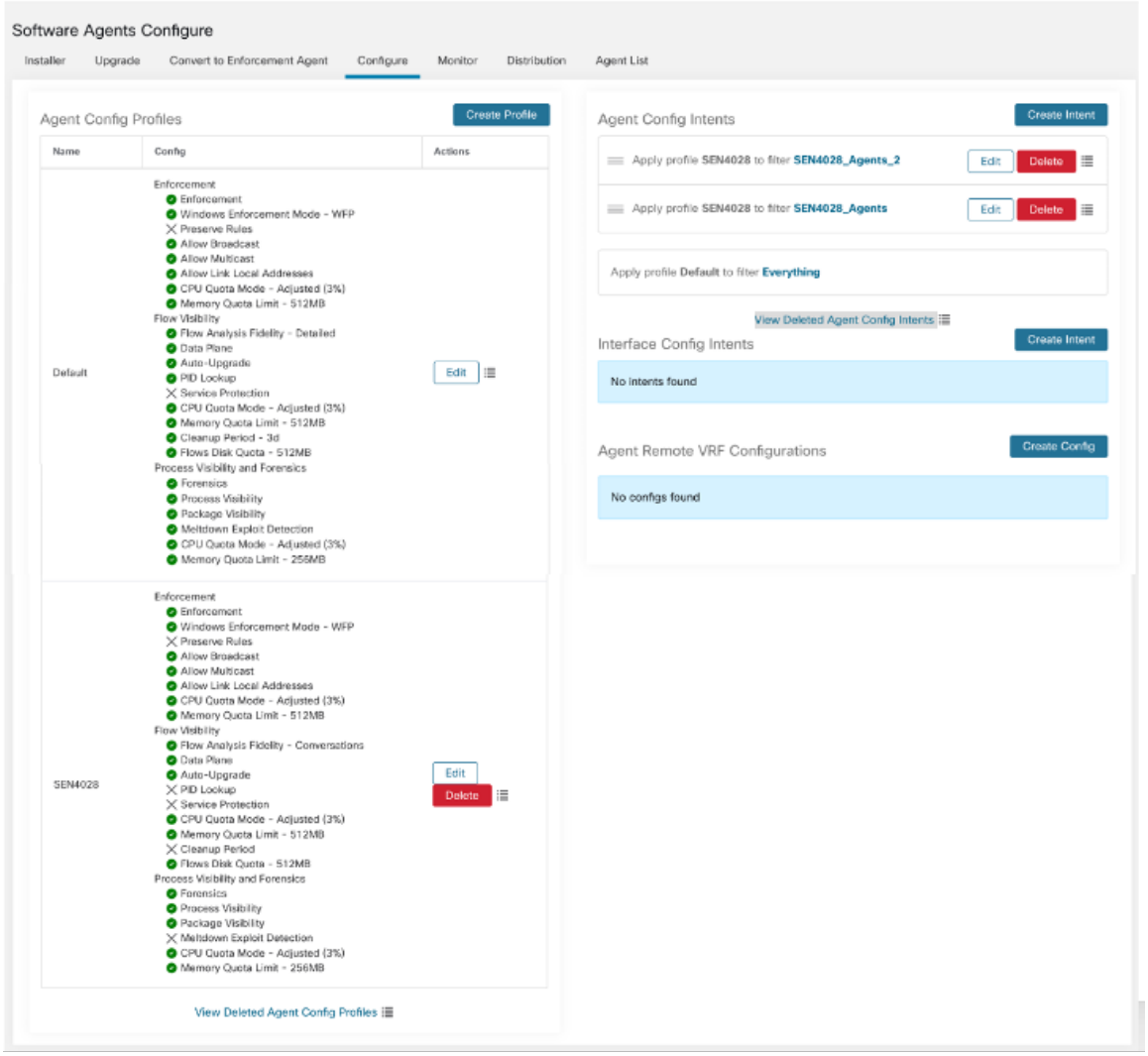


ソフトウェアエージェントの構成

ソフトウェアエージェントは、**エージェント構成プロファイル**を**インベントリフィルタ**または**範囲**に関連付ける**エージェント構成インテント**を作成することによって構成されます。詳細については、「インベントリ （407ページ）」の章を参照してください。最初に一致したインテントが各エージェントに適用されます。Secure Workload展開には、特定の構成プロファイルに関連付けられていないすべてのエージェントに適用される、デフォルトのエージェント構成が常に存在します。

図 *11* : ソフトウェアエージェントの設定





エージェント設定プロファイルの作成

始める前に

ソフトウェアエージェント設定の要件と前提条件 （87 ページ）を参照してください。

手順

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]**>**[エージェント（Agents）]**をクリックします。

ステップ 2 [Configure] タブをクリックします。

ステップ 3 [プロファイルの作成（Create Profile）] ボタンをクリックします。

ステップ 4 プロファイルの名前を入力し（必須）、プロファイルを使用できる範囲を選択します。

ステップ 5 次の表にリストされているフィールドに適切な値を入力します。

表 *12* : 適用設定

オプション	説明
施行	[有効（Enable）]：エージェントでのポリシー適用を有効にします。 適用を有効にすると、エージェントは最後に受信したポリシーセットをすぐに適用します（存在する場合）。 [無効（Disable）]（デフォルト）：エージェントはポリシーを適用しません。 （注） 有効化された適用を無効にしてから再度有効にすると、ファイアウォールの状態がクリアされ、キャッチオールのデフォルトアクションが ALLOW に設定されます。
[ルールの保持（Preserve Rules）]	[有効（Enable）]：エージェントの既存のファイアウォールルールを保持します。 [無効（Disable）]（デフォルト）：Secure Workload から適用ポリシールールを適用する前に、既存のファイアウォールルールをクリアします。 動作はプラットフォームによって異なります。各プラットフォームの詳細を確認するには、このドキュメントで「ルールの保持」を検索してください。
[ブロードキャストの許可（Allow Broadcast）]	[有効（Enable）]（デフォルト）：ファイアウォールにルールを追加して、ワークロードの入力および出力ブロードキャスト トラフィックを許可します。 [無効（Disable）]：ルールを追加しません。エージェントのデフォルトポリシーが deny である場合、ブロードキャストトラフィックはドロップされます。
[マルチキャストの許可（Allow Multi-casting）]	[有効（Enable）]（デフォルト）：ファイアウォールにルールを追加して、ワークロードの入力および出力マルチキャストトラフィックを許可します。 [無効（Disable）]：ルールを追加しません。エージェントのデフォルトポリシーが deny である場合、マルチキャストトラフィックはドロップされます。
[リンクローカルの許可（Allow Link Local）]	[有効（Enable）]（デフォルト）：ファイアウォールにルールを追加して、ワークロードのリンクローカルアドレスのトラフィックを許可します。 [無効（Disable）]：ルールを追加しません。エージェントのデフォルトポリシーが deny である場合、マルチキャストトラフィックはドロップされます。

オプション	説明
適用プロセスの CPU クォータモード	[調整済み（Adjusted）]（デフォルト）：CPU 制限は、システム上の CPU の数に応じて調整されます。たとえば、CPU制限が3%に設定されていて、システムに 10 個の CPU がある場合、このモードを選択すると、エージェントは合計 30%（上位で測定）の使用が許可されます。 [上位（Top）]：CPU制限値は、平均上位ビューと一致します。たとえば、CPU 制限が 3% に設定されていて、システムに 10 個の CPU があるとします。この場合でも CPU 使用率は 3% のままとなります。これはかなり制限の厳しいモードですので、必要な場合にのみ使用してください。 [無効（Disable）]：CPU 制限機能が無効になります。エージェントは、OS で許可されている CPU リソースを使用します。 詳細については、agent_cpu_sla.pdf を参照してください。
[CPUクォータ制限（%）（CPU Quota Limit (%)）]	エージェントが使用できるシステム処理能力の実際の制限をパーセントで指定します。
[メモリクォータ制限（MB）（Memory Quota Limit (MB)）]	プロセスが使用できるメモリ制限を MB 単位で指定します。プロセスがこの制限に達すると、再起動します。
[Windows適用モード（Windows Enforcement Mode）]	Windows ワークロードでは、エージェントは以下を使用してネットワークポリシーを適用できます。 • [WFP]：Windows Filtering Platform（Windows Filter Engine で WFP フィルタを直接プログラミングします）。 WFP モードでの Windows プラットフォームにエージェントを適用（67 ページ）を参照してください。 • [WAF]（デフォルト）：Windows Advanced Firewall。 WAF モードの Windows プラットフォームでのエージェントの適用（64 ページ）を参照してください。

表 *13 :* フローの可視性設定

フィールド	説明
[流動解析の忠実度（Flow Analysis Fidelity）]	[カンバセーション（Conversations）] : すべてのセンサーでカンバセーションモードを有効にします。 [詳細（Detailed）]（デフォルト）：すべてのセンサーで詳細モードを有効にします。 .
Data Plane	[有効（Enable）]（デフォルト）：エージェントがクラスタにレポートを送信できるようにします。 [無効（Disable）] : エージェントのレポートを無効にします。
自動アップグレード	[有効（Enable）]（デフォルト）：新しいパッケージが利用可能になったときに、エージェントを自動的にアップグレードします。 [無効（Disable）] : エージェントを自動的にアップグレードしません。
[PIDルックアップ（PID Lookup）]	[有効（Enable）] : エージェントでPIDルックアップを有効にします。有効にすると、エージェントは、ネットワークフローをワークロードで実行中のプロセスに関連付けるようベストエフォートで試みます。この操作はコストがかかる可能性があるため、エージェントは各エクスポートサイクルで実行される操作の数を抑制して、CPU オーバーヘッドを制御します。設定が有効になっていても、一部のフローがどのプロセスにも関連付けられていない場合があります。 [無効（Disable）]（デフォルト）：エージェントで PID ルックアップを有効にしません。

フィールド	説明
サービスの保護（Service Protection）	[有効（Enable）]：エージェントでサービスの保護を有効にします。有効にすると、ユーザーによるサービスの無効化、エージェントのアンインストール、およびサービスの再起動が、エージェントにより防がれます。ただし、サービスの保護を無効にすると、ユーザーは引き続きエージェントの停止またはアンインストールを行うことができます。 （注） • エージェントの通常の自動アップグレードを行う場合は、サービスの保護を無効にしないでください。 • エージェントの手動アップグレードを行う場合は、サービスの保護を有効にしないでください。 • サービスの保護は、強制アップグレードをブロックします（インストーラスクリプトの -forceUpgrade オプションの使用など）。 • サービスの保護が有効になっている場合、システムによって開始されたアップグレードはシームレスに機能します。 [無効（Disable）]（*）：デフォルトでは、エージェントのサービスの保護は無効になっています。 [詳細（Detailed）]（デフォルト）：すべてのセンサーで詳細モードを有効にします。 （注） この機能は、Windows エージェントでのみ使用できます。
[CPUクォータモード（CPU Quota Mode）]	[調整済み（Adjusted）]（デフォルト）：CPU 制限は、システム上の CPU の数に応じて調整されます。たとえば、CPU制限が3%に設定されていて、システムに 10 個の CPU がある場合、このモードを選択すると、エージェントは合計 30%（上位で測定）の使用が許可されます。 [上位（Top）]：CPU制限値は、平均上位ビューと一致します。たとえば、CPU 制限が 3% に設定されていて、システムに 10 個の CPU があるとします。この場合でも CPU 使用率は 3% のままとなります。これはかなり制限の厳しいモードですので、必要な場合にのみ使用してください。 [無効（Disable）]：CPU 制限機能が無効になります。エージェントは、OS で許可されている CPU リソースを使用します。 詳細については、agent_cpu_sla.pdf を参照してください。
[CPUクォータ制限（%）（CPU Quota Limit (%)）]	エージェントが使用できるシステム処理能力の実際の制限をパーセントで指定します。

フィールド	説明
[メモリクォータ制限（MB）（Memory Quota Limit (MB)）]	プロセスが使用できるメモリ制限を MB 単位で指定します。プロセスがこの制限に達すると、再起動します。
[クリーンアップ期間（日）（Cleanup period (days)）]	[有効（Enable）]：エージェントの自動クリーンアップを有効にします。非アクティブなエージェントを削除するまでの日数を入力します。 [無効（Disable）]（デフォルト）：エージェントで自動クリーンアップを有効にしません。
[流動解析の忠実度（Flow Analysis Fidelity）]	[カンバセーション（Conversations）]：すべてのセンサーでカンバセーションモードを有効にします。 [詳細（Detailed）]（デフォルト）：すべてのセンサーで詳細モードを有効にします。
フローのディスククォータ（MB）（Flows Disk Quota (MB)）	保存されるフローデータの合計サイズ制限を MB 単位で指定します。
フローの時間枠（h）（Flows Time Window (h)）	エージェントがフローをキャプチャしてローカルに保存する必要がある期間を時間単位で指定します。 [フローのディスククォータ（Flows Disk Quota）] または [フローの時間枠（Flows Time Window）]（サイズベースまたは時間ベースのローテーション）を選択できます。[フローの時間枠（Flows Time Window）] を選択すると、[フローのディスククォータ（Flows Disk Quota）] は 16GB に設定されます。[フローのディスククォータ（Flows Disk Quota）] を 0 に設定すると、この機能は無効になります。フローデータは、サイズ制限または時間制限に達するとローテーションされます。

図 *12 :* フローの可視性

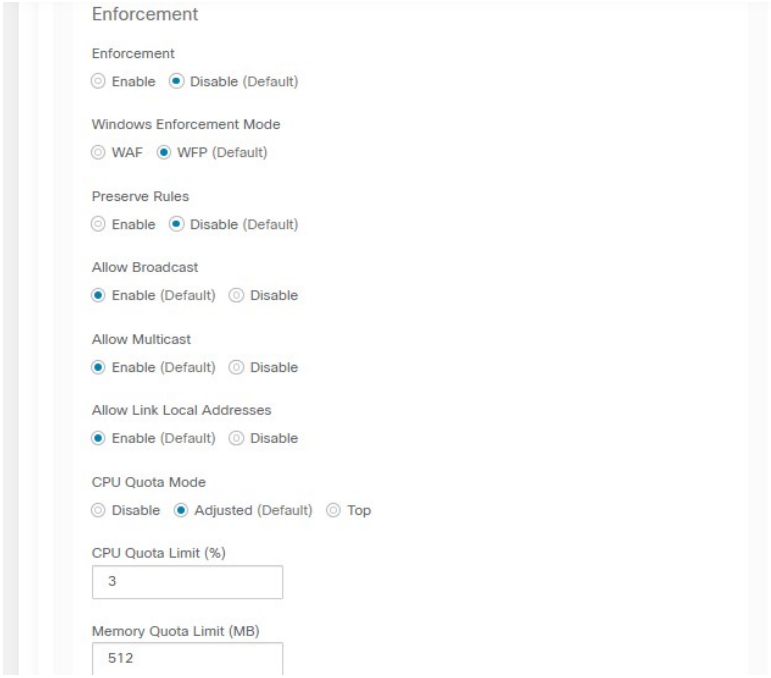

Enforcement
Enforcement
Enable
Disable (Default)
Windows Enforcement Mode
WAF
WFP (Default)
Preserve Rules
Enable
Disable (Default)
Allow Broadcast
Enable (Default)
Disable
Allow Multicast
Enable (Default)
Disable
Allow Link Local Addresses
Enable (Default)
Disable
CPU Quota Mode
Disable
Adjusted (Default)
Top
CPU Quota Limit (%)
3
Memory Quota Limit (MB)
512

図 *13 :*

表 *14*：プロセスの可視性とフォレンジック設定

フィールド	説明
フォレンジック	[有効（Enable）]：エージェントでフォレンジックを有効にします。この機能は、次の CPU 制限で指定された追加の CPU サイクルを消費する可能性があることに注意してください。たとえば、CPU 制限が 3% で、この機能が有効になっている場合、エージェントは合計で最大 6% を使用できると想定します。 [無効（Disable）]（デフォルト）：エージェントでフォレンジックを無効にします。
[Meltdown エクスプロイト検出（Meltdown Exploit Detection）]	[有効（Enable）]：エージェントで Meltdown エクスプロイト検出を有効にします。この機能を使用するには、フォレンジックを有効にする必要があります。詳細については、「互換性（698ページ）」の「サイドチャネル」を参照してください。 [無効（Disable）]（デフォルト）：エージェントで Meltdown エクスプロイト検出を無効にします。
[CPUクォータモード（CPU Quota Mode）]	[調整済み（Adjusted）]（デフォルト）：CPU 制限は、システム上の CPU の数に応じて調整されます。たとえば、CPU制限が3%に設定されていて、システムに 10 個の CPU がある場合、このモードを選択すると、エージェントは合計 30%（**上位**で測定）の使用が許可されます。 [上位（Top）]：CPU制限値は、平均上位ビューと一致します。たとえば、CPU 制限が 3% に設定されていて、システムに 10 個の CPU があるとします。この場合でも CPU 使用率は 3% のままとなります。これはかなり制限の厳しいモードですので、必要な場合にのみ使用してください。 [無効（Disable）]：CPU 制限機能が無効になります。エージェントは、OS で許可されている CPU リソースを使用します。
[CPUクォータ制限（%）（CPU Quota Limit (%)）]	エージェントが使用できるシステム処理能力の実際の制限をパーセントで指定します。
[メモリクォータ制限（MB）（Memory Quota Limit (MB)）]	プロセスが使用できるメモリ制限を MB 単位で指定します。プロセスがこの制限に達すると、再起動します。

ステップ 6 [保存（Save）] をクリックします。

次のタスク

このプロファイルをエージェント設定インテントに関連付けます。エージェント構成インテントの作成 （99 ページ） を参照してください。

エージェント構成インテントの作成

始める前に

- ソフトウェアエージェント設定の要件と前提条件 （87 ページ）を参照してください。
- エージェント設定プロファイルを作成します。エージェント設定プロファイルの作成 （90 ページ）を参照してください。

手順

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]**>**[エージェント（Agents）]**をクリックします。

ステップ 2 [Configure] タブをクリックします。

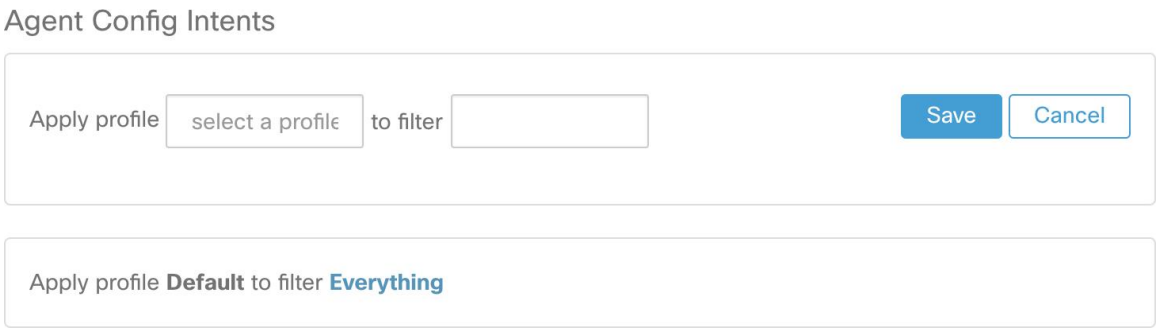
ステップ 3 [エージェント構成インテント（Agent Config Intent）] 見出しの横にある [インテントの作成（Create Intent）] ボタンをクリックします。

ステップ 4 次の表にリストされているフィールドに適切な値を入力します。

フィールド	説明
[プロファイル（Profile）]（必須）	既存のプロファイルの名前を入力し、ドロップダウンメニューからそのプロファイル名を選択します。
[フィルタ（Filter）]（必須）	既存のフィルタまたは範囲の名前を入力するか、ドロップダウンメニューから [新しいフィルタの作成（Create new filter）] を選択します。 フィルタの作成の詳細については、「フィルタ」を参照してください。

ステップ 5 [保存（Save）] をクリックします。

図 *14* : エージェント構成インテント



エージェントのリモート VRF 設定の作成

これは、Secure Workload ソフトウェアエージェントに VRF を割り当てる際に推奨される方法です。Secure Workload アプライアンスは、この設定を使用して、Secure Workload アプライアンスへの接続時にエージェントに表示されるソース IP アドレスとソースポートに基づいて、VRF をソフトウェアセンサーに割り当てます。

手順

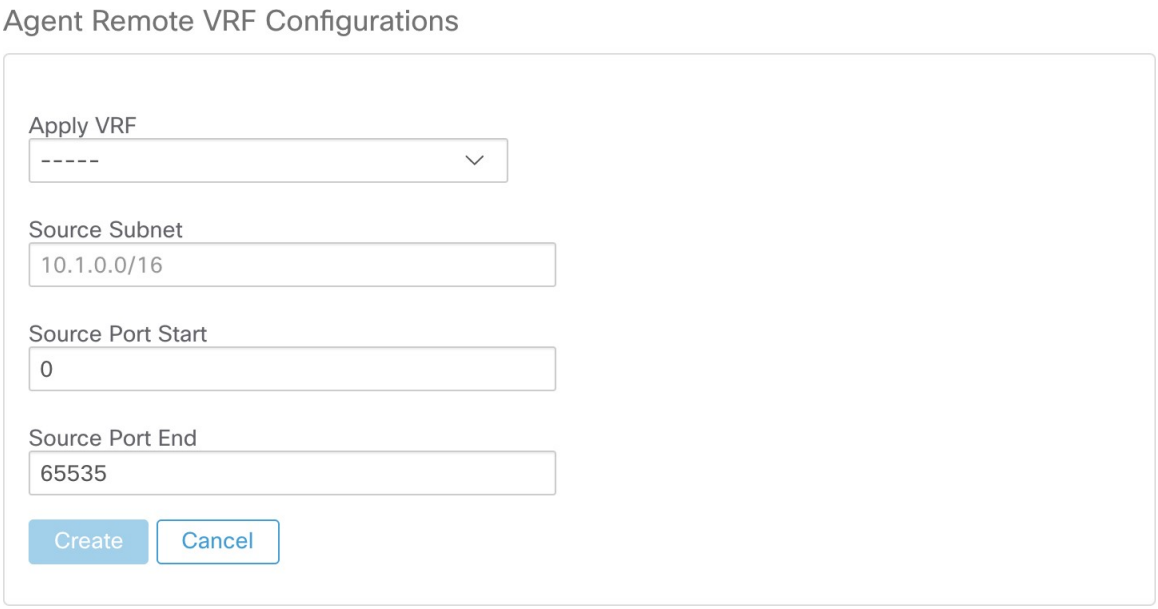
ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]**>**[エージェント（Agents）]**をクリックします。

ステップ 2 [Configure] タブをクリックします。

ステップ 3 [エージェントのリモートVRF設定（Agent Remote VRF Configurations）] 見出しの横にある [設定の作成（Create Config）] ボタンをクリックします。

ステップ 4 フィールドに適切な値を入力し、[保存（Save）] をクリックします。

図 *15* : リモート *VRF* 設定



インターフェイス構成インテントの作成

VRF をエージェントに割り当てる推奨される方法は、リモート VRF 構成設定を使用する方法です。まれなケースですが、エージェントホストに複数のインターフェイスがあり、異なる VRF を割り当てる必要がある場合、ユーザーはインターフェイス構成インテントを使用してインターフェイスに VRF を割り当てることができます。

手順

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]**>**[エージェント（Agents）]** をクリックします。

ステップ 2 [Configure] タブをクリックします。

ステップ 3 [インターフェイス構成インテント（Interface Config Intent）] 見出しの横にある [インテントの作成（Create Intent）] ボタンをクリックします。

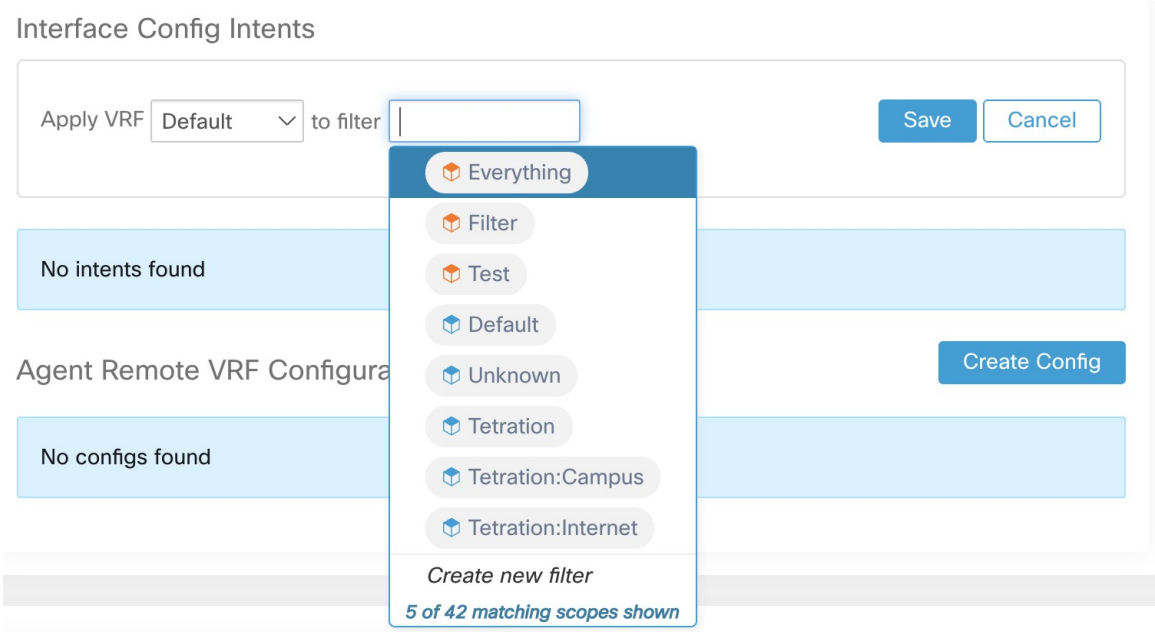
ステップ 4 次の表にリストされているフィールドに適切な値を入力します。

フィールド	説明
VRF	（必須）ドロップダウンメニューから VRF を選択します。

フィールド	説明
フィルタ	（必須）既存のフィルタまたは範囲の名前を入力するか、ドロップダウンメニューから[新しいフィルタの作成（Create new filter）] を選択します。 フィルタの作成の詳細については、「フィルタ」を参照してください。

ステップ 5 [保存（Save）] をクリックします。

図 *16*：インターフェイス設定インテント



（注） Catch All インターフェイス構成インテントが適用されないという既知の問題があります。これは、より優先順位の高いインターフェイス構成インテントをユーザーが削除する場合にのみ当てはまります。そのような場合、エージェントはデフォルトの Catch All インテントにフォールバックしません。

ワークロードプロファイルでの詳細なエージェントステータスの表示

手順

ステップ 1 上記の手順に従って、エージェントのステータスを確認します。

ステップ 2 [適用エージェント（Enforcement Agents）] ページで、[エージェントOS分布（Agent OS Distribution）] をクリックします。OS を選択し、ボックスの右上隅にあるフィルタイメージをクリックします。

ステップ 3 [ソフトウェアエージェントリスト（Software Agents Agent List）] ページに、エージェントと選択した OS の分布が一覧表示されます。

ステップ 4 [エージェント（Agent）] をクリックすると、[エージェントの詳細（Agent Details）] セクションが表示されます。IPアドレスをクリックして、[ワークロードプロファイル（Workload Profile）] ページに移動します。

ステップ 5 [ワークロードプロファイル（Workload Profile）] ページでは、ホストプロファイル、エージェントプロファイルや、帯域幅、長期プロセス、パッケージ、プロセススナップショット、設定、インターフェイス、統計、ポリシー、コンテナポリシーなど、他のエージェント固有の詳細情報が表示されます。

ステップ 6 [設定（Config）] タブをクリックして、エンドホストの設定を表示します。

ステップ 7 [ポリシー（Policies）] タブをクリックして、エンドホストに適用されたポリシーを表示します。

図 *17* : ワークロードプロファイル - 設定

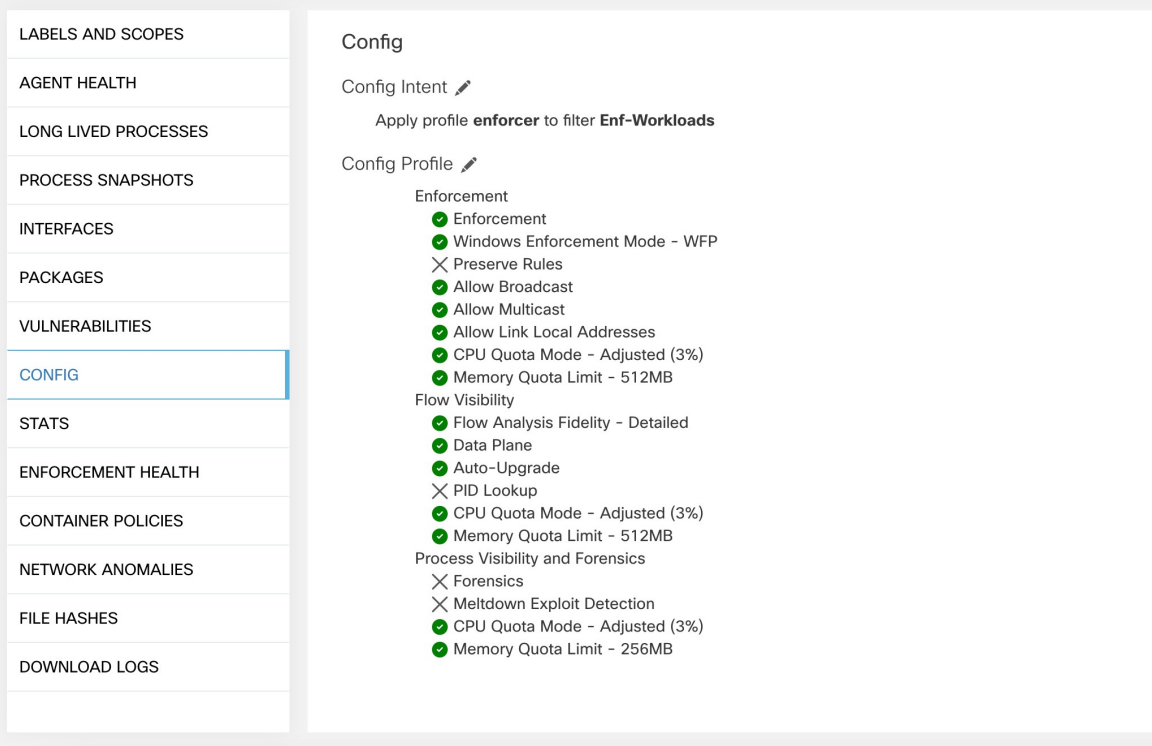
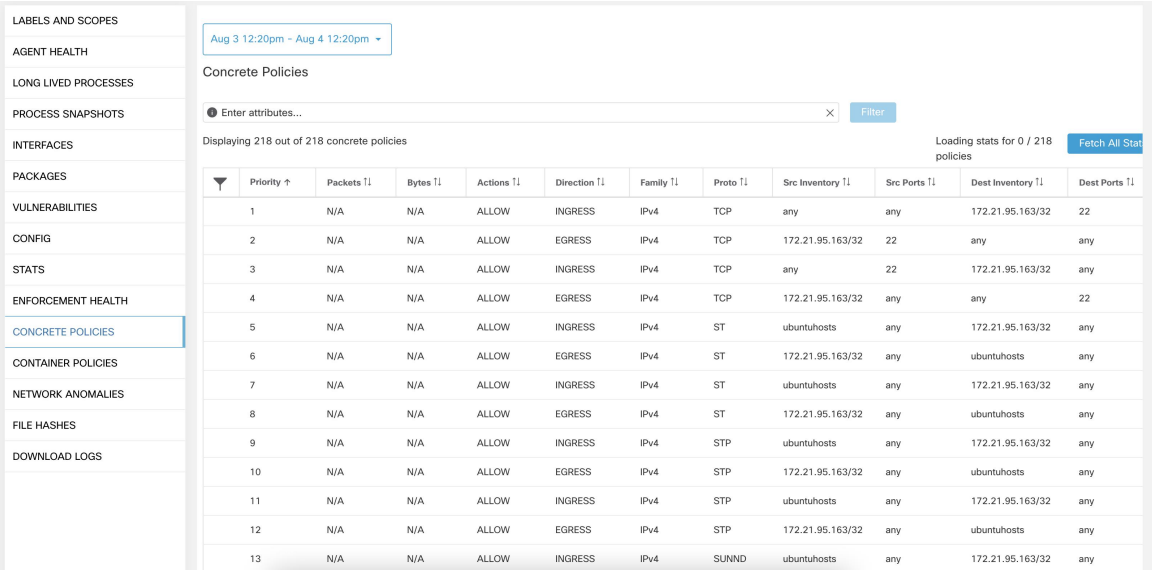


図 *18* : ワークロードプロファイル - ポリシー



エージェントのリホーム

エージェントのリホームは、お客様をオンプレミスから TaaS に、または TaaS からオンプレミスに移行するために使用されます。

ユーザロール

- サイト管理者
- カスタマーサポート

TaaS に関連する移行、特に TaaS からオンプレミスへの移行を行うお客様は、内部サポートチームと連携する必要があります。

ワークフロー

- アクティベーションキー、センサー VIP、およびセンサー CA 証明書を入力し、リホームの有効化 （105 ページ）を行います。
- リホームするエージェントの選択 （107 ページ）。
- リホームの無効化 （108 ページ）。

（注） リホームは、クラスタまたはアプライアンス全体のアクティビティです。一度に移動できる宛先は 1 つだけです。完了したら、エージェントのリホームを無効にすることを推奨します。

リホームの有効化

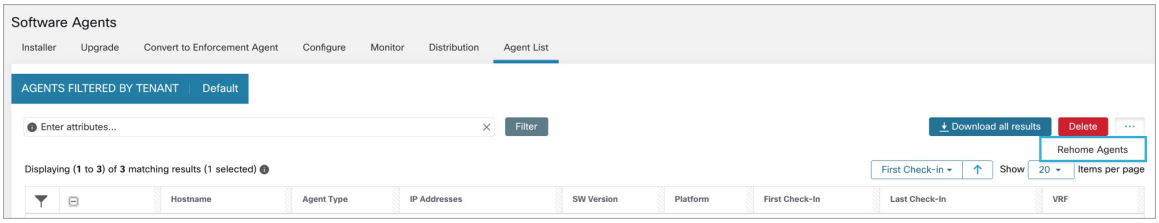
手順

ステップ 1 左側のナビゲーションメニューで、[管理（Manage）] > [ワークロード（Workloads）] > [エージェント（Agents）] をクリックします。

ステップ 2 [エージェントリスト（Agent List）] タブをクリックします。

ステップ 3 メニューアイコンをクリックし、[エージェントのリホーム（Rehome Agents）] を選択します。

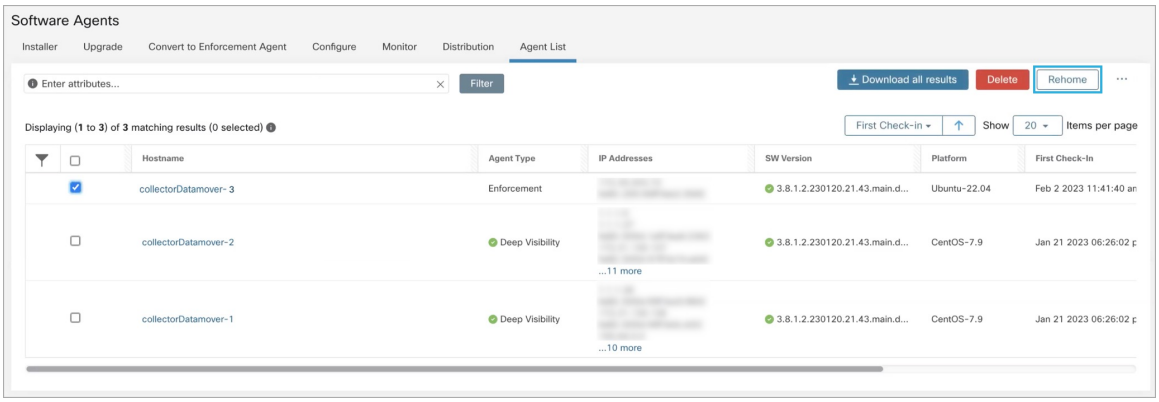
図 *19 :* エージェントのリホーム（*Rehome Agents*）



ステップ 4 [エージェントのリホーム（Agent Rehoming）] ウィンドウで、次の詳細を入力します。

フィールド	説明
宛先範囲のアクティベーションキー（Destination Scope Activation Key）	**1.** [管理（Manage）] > [ワークロード（Workloads）]>[エージェント（Agents）] に移動します。 **2.** [インストーラ（Installer）] タブをクリックします。 **3.** [従来のパッケージインストーラを使用した手動インストール（Manual install using classic packaged installers）] を選択します。 **4.** [次へ（Next）] をクリックします。 **5.** [エージェントのアクティベーションキー（Agent Activation Key）] をクリックします。 **6.** [キー（Key）] の値をコピーし、[宛先範囲のアクティベーションキー（Destination Scope Activation Key）] フィールドに貼り付けます。
宛先センサーVIP（Destination Sensor VIP）	**1.** [プラットフォーム（Platforms）]>[クラスタ設定（Cluster Configuration）] に移動します。 **2.** [センサーVIP（Sensor VIP）] をコピーし、[宛先センサーVIP（Destination Sensor VIP）] フィールドに貼り付けます。
HTTPS プロキシ	アウトバウンド通信にプロキシを使用するためにエージェントで必要となる場合は、 プロキシドメインまたはアドレスを入力します。
宛先センサーCA証明書（Destination Sensor CA Cert）	**1.** [プラットフォーム（Platforms）]>[クラスタ設定（Cluster Configuration）] に移動します。 **2.** [センサーCA証明書のダウンロード（Download Sensor CA Cert）] をクリックします。

図 *20* : エージェントのリホームの有効化



ステップ 5 [エージェントのリホームの有効化（Enable Agent Rehomeing）] をクリックします。

設定が保存されます。[リホーム（Rehome）] ボタンが右上に表示されます。

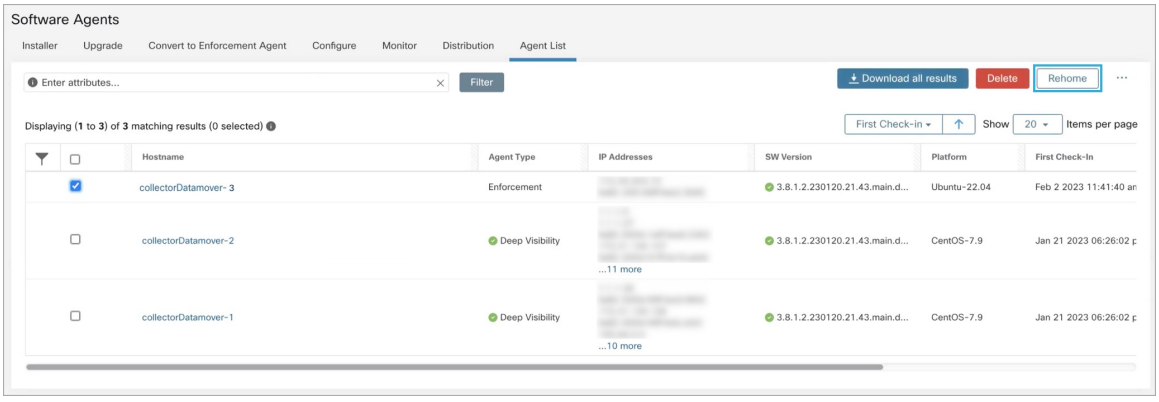
リホームするエージェントの選択

手順

ステップ 1 エージェントを選択します。

ステップ 2 [リホーム（Rehome）] をクリックします。

図 *21* : リホームするエージェントの選択



ステップ 3 確認のために [はい（Yes）] をクリックします。

リホームの無効化

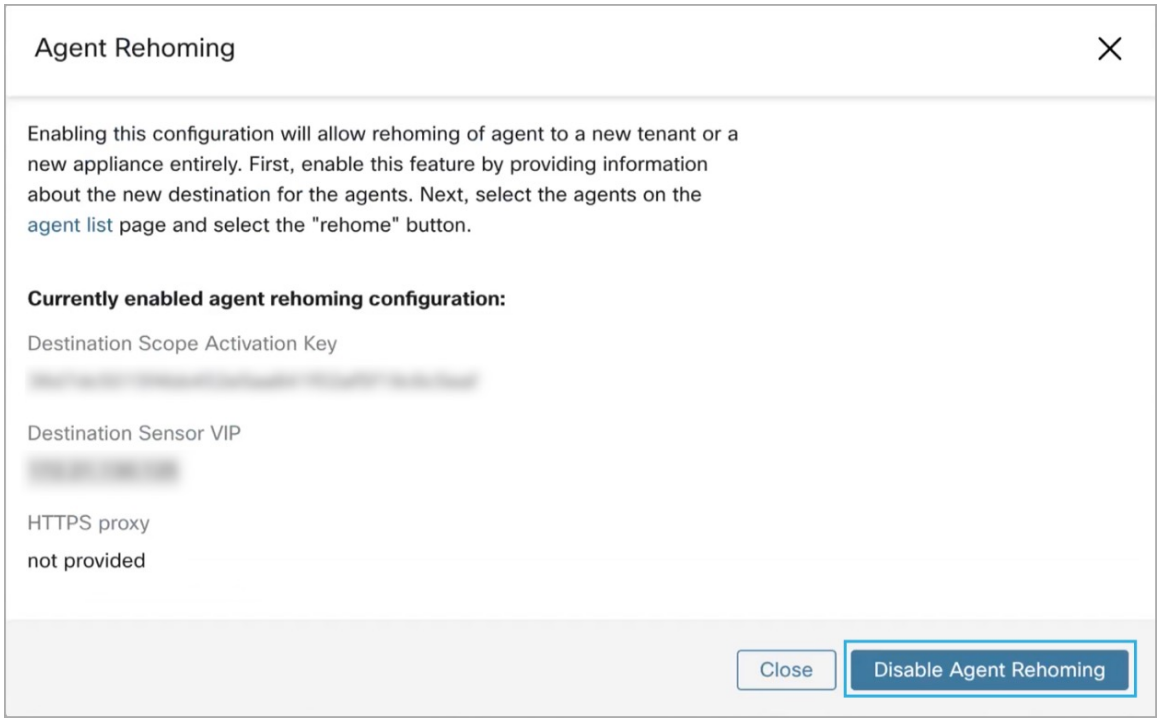
（注） TaaS との間で複数のお客様がリホームする場合、サイト管理者はそれぞれのお客様を個別に移動する必要があります。つまり、リホームを無効にしてすべての設定をクリアし、新しいお客様に対してリホームを有効にします。

手順

ステップ 1 メニューアイコンをクリックし、[エージェントのリホーム（Rehome Agents）] を選択します。

ステップ 2 [エージェントのリホーム（Agent Rehomeing）] ウィンドウで、[エージェントのリホームの無効化（Disable Agent Rehomeing）] をクリックします。

図 *22 :* エージェントのリホームの無効化（***Disable Agent Rehoming***）



適用が有効である場合のホスト IP アドレスの変更

適用が有効になっている場合にホストの IP アドレスを変更すると、ホスト IP がホスト ファイアウォールルールで認識され、Catch All が拒否に設定されている場合に影響が出る可能性があります。このシナリオでは、次の手順でホスト IP アドレスを変更することをお勧めします。

手順

ステップ 1 Secure Workload UI で、適用を無効にして新しいエージェント構成プロファイルを作成します。

ステップ 2 IP アドレスを変更する必要があるホスト、およびその古い IP アドレスと新しい IP アドレスのリストを含むインテントを作成します。

ステップ 3 新しく作成したエージェント構成プロファイルをこのインテントに適用し、インテントを保存します。

ステップ 4 これらの選択されたホストでは、適用を無効にしておく必要があります。

ステップ 5 これらのホストの IP アドレスを変更します。

ステップ 6 Secure Workload UI で、範囲内のフィルタをこれらのホストの新しい IP アドレスで更新します。

ステップ 7 [エージェントワークロードプロファイル（Agent Workload Profile）] ページの [インターフェイス（Interfaces）] タブで IP アドレスの変更を確認します。[ポリシー（Policies）] タブで、ポリシーが新しい IP アドレスで生成されていることを確認します。

ステップ 8 上記で作成したインテントおよびプロファイルを削除します。

ステップ 9 範囲の元のエージェント構成プロファイルで適用が無効になっている場合は、適用を有効にします。

ソフトウェアエージェントのアップグレード

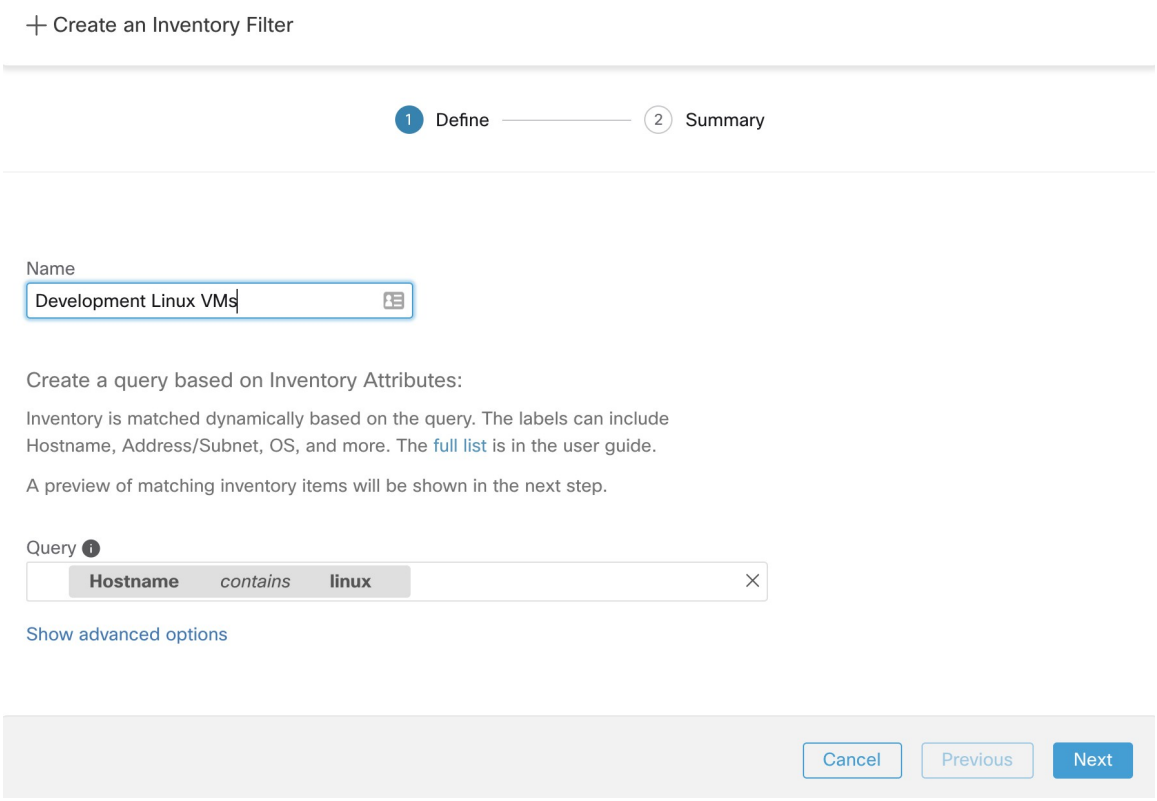
UI からのエージェントのアップグレード

エージェントは、「ソフトウェアエージェントの設定」で詳しく説明されているエージェント設定インテントワークフローを使用してアップグレードできます。エージェント設定プロファイルの設定時に、[有効（Enabled）] または [無効（Disabled）] にできる [自動アップグレード（Auto Upgrade）] オプションがあります。オプションが [有効（Enabled）] になっている場合、インベントリフィルタ条件に一致するエージェントは、利用可能な最新バージョンのソフトウェアに自動的にアップグレードされます。

次のセクションでは、ソフトウェアエージェント設定インテントワークフローを使用して、ソフトウェアエージェントのアップグレード動作を指定する方法について説明します。

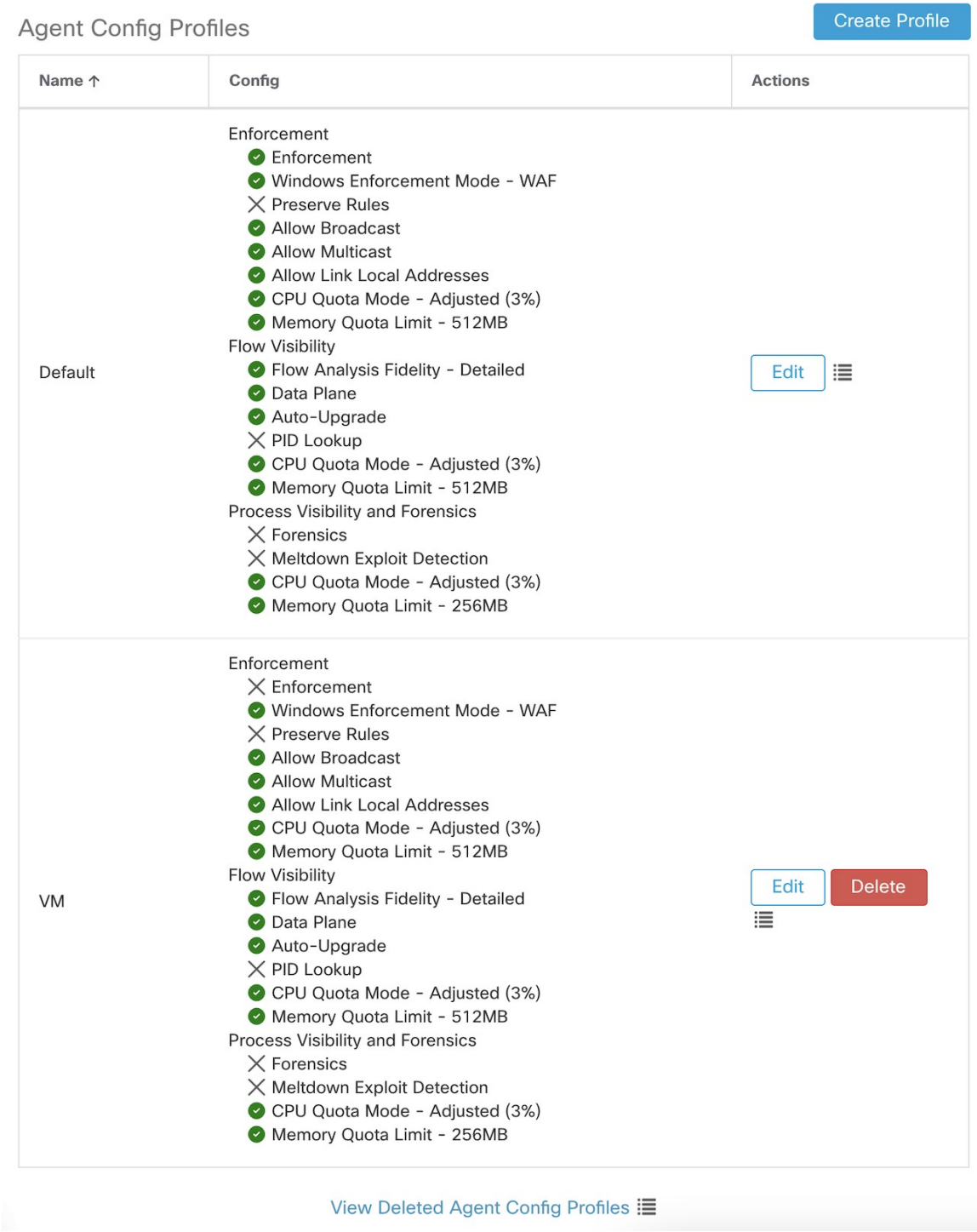
1. [インベントリフィルタ（Inventory Filters）] ページでインベントリフィルタを作成します。詳細については「フィルタ」をご覧ください。

図 *23* : インベントリフィルタ



2. 上記のインベントリフィルタによって選択されたエージェントにユーザーが適用するエージェント設定プロファイルを作成します。エージェント設定プロファイルには、選択したエージェントを自動アップグレードするかどうかを制御する [自動アップグレード（Auto Upgrade）] オプションがあることに注意してください。

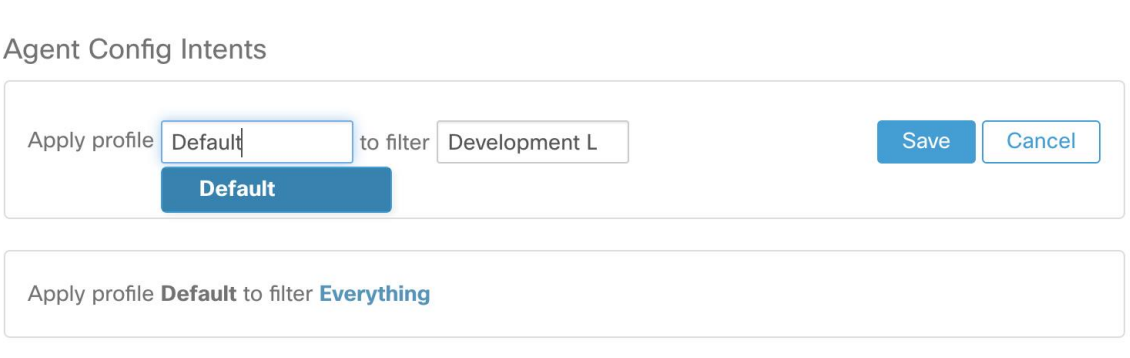
図 *24* : エージェント設定



3. 最後に、選択した設定プロファイルを、選択したエージェントのセットに適用するエージェント設定インテントを作成する必要があります（インベントリフィルタを使用）。自動アップグレードオプションが有効になっている場合、選択したすべてのエージェントが

自動アップグレードされます。通常、エージェントプロファイルがエージェントに適用された後、エージェントがアップグレードされるまでに最大 30 分かかることがあります。

図 *25 :* エージェント設定インテント



（注） デフォルトのエージェントプロファイルの自動アップグレード設定は、ERSPAN に適用されます。

次のセクションでは、センサー設定インテントワークフローを使用せずに、エージェントを手動でアップグレードする方法について説明します。

手順

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]>[エージェント（Agents）]** をクリックします。

ステップ 2 [アップグレード（Upgrade）] タブをクリックします。

ステップ 3 詳細可視性エージェントと適用エージェントのみが表示され、エージェントごとに、アップグレード可能な新しいバージョンのみがリストに表示されます。デフォルトでは、最新バージョンが選択されます。

ステップ 4 フィルタボックスに検索クエリを入力して、エージェントリストをフィルタリングします。たとえば、「`Platform = CentOS-7.6`」と入力します。

ステップ 5 このバージョンにアップグレードするエージェントを選択し、[アップグレード（Upgrade）] ボタンをクリックします。

（注） 通常の状況では、エージェントにアップグレードを処理させることが強く推奨されます。これがサポートされている唯一のアップグレード方法です。ユーザーが新しいバージョンを手動でダウンロードして実行中のエージェントに直接適用する仕方でアップグレードを制御する場合は、安全上の注意事項に従いつつ実行する必要があります。

Kubernetes/Openshift エージェントの動作のアップグレード

daemonset インストーラスクリプトを使用して Kubernetes/Openshift ノードにインストールされたエージェントは、セルフアップグレードが可能です。アップグレードプロセスは、自動アップグレードオプションによって、または Kubernetes/Openshift クラスタ内の任意のノードに対してアップグレードを手動でトリガーすることによって制御されます。この環境でのアップグレードのメカニズムでは、daemonset 仕様で Docker イメージをアップグレードします。つまり、次の段落で説明するように、1 つのエージェントのアップグレードが、daemonset によってカバーされるすべてのエージェントに影響します。

Daemonset Pod 仕様が変更されると、Kubernetes/Openshift はグレースフルシャットダウンをトリガーし、新しい Docker イメージを取得して、Kubernetes/Openshift クラスタ内のすべてのノードで Secure Workload エージェントポッドを開始します。これにより、アップグレードを許可するポリシーがクラスタ内のノードのサブセットにのみ適用される場合でも、エージェントは他のノードでアップグレードされます。

すべてのノードで自動アップグレードが無効になっている場合は、新しいインストーラスクリプトをダウンロードしてインストールを再実行することにより、手動アップグレードが可能です。インストールスクリプトは、新しいインストールと既存のインストールのアップグレードの場合を自動検出し、インストール済みであることを検出すると、daemonset ポッドを手動でアップグレードするように機能します。

ソフトウェアエージェントの削除

優れた可視性または適用 Linux エージェントの削除

RPM ベースのインストール：

1. コマンド **rpm -e tet-sensor** を実行します。

エージェントのアンインストールイベントがクラスタに通知され、[ソフトウェアエージェント（Software Agent）] ページでエージェントがアンインストール済みとしてマークされます。

[ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを手動で削除します。また、ユーザーは、エージェント設定プロファイルから [クリーンアップ期間（cleanup period）] をオンにして、エージェントの自動クリーンアップまたは削除を有効にすることもできます。



（注） デフォルトでは、[クリーンアップ期間（cleanup period）] はオフになっています。

Ubuntu.deb ベースのインストール：

Ubuntu エージェントの新規インストールでは、ネイティブの .deb 形式が使用されるようになりました。

1. コマンド **dpkg –purge tet-sensor** を実行します。

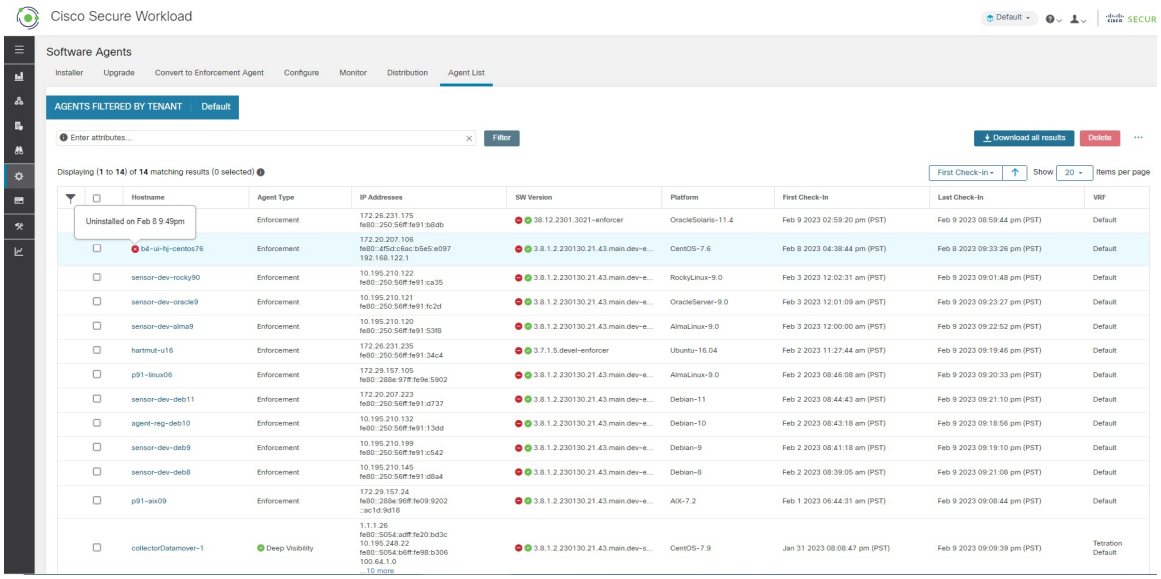
エージェントのアンインストールイベントがクラスタに通知され、[ソフトウェアエージェント（Software Agent）] ページでエージェントがアンインストール済みとしてマークされます。

[ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを手動で削除します。また、ユーザーは、エージェント設定プロファイルから [クリーンアップ期間（cleanup period）] をオンにして、エージェントの自動クリーンアップまたは削除を有効にすることもできます。

（注）

- デフォルトでは、[クリーンアップ期間（cleanup period）] はオフになっています。
- エージェントの操作中に、一部のカーネルモジュールがカーネルによって自動的にロードされる可能性があります。たとえば、Linux で適用が有効になっている場合、Netfilter モジュールがロードされる可能性があります。エージェントには、カーネルによってロードされたモジュールのリストがありません。したがって、エージェントのアンインストール中に、カーネルモジュールをアンロードできない可能性があります。
- 適用エージェントがシステムファイアウォールにポリシーを適用した場合、エージェントをアンインストールすると、適用されたポリシーがクリアされ、システムファイアウォールが開きます。

図 *26 :* エージェントのアンインストールアラート



優れた可視性/適用 Widnows エージェントの削除

Secure Workload エージェントをアンインストールするには、次の 2 つのオプションがあります。

手順

ステップ 1 [コントロールパネル（Control Panel）] → [プログラム（Programs）] → [プログラムと機能（Programs And Features）] に移動し、**Cisco Secure Workload Agent**（Cisco Tetration Agent）をアンインストールします。

ステップ 2 または、「**C:\Program Files\Cisco Tetration**」内のショートカット **Uninstall.lnk** を実行します。

ステップ 3 適用エージェントがシステムファイアウォールにポリシーを適用した場合、エージェントをアンインストールすると、適用されたポリシーがクリアされ、システムファイアウォールが開きます。

エージェントのアンインストールイベントがクラスタに通知され、[ソフトウェアエージェント（Software Agent）] ページでエージェントがアンインストール済みとしてマークされます。

[ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを手動で削除します。また、ユーザーは、エージェント設定プロファイルから [クリーンアップ期間（cleanup period）] をオンにして、エージェントの自動クリーンアップまたは削除を有効にすることもできます。

（注） デフォルトでは、[クリーンアップ期間（cleanup period）] はオフになっています。

（注）
- エージェントのインストール中に Npcap がインストールされている場合、それもアンインストールされます。
- デフォルトのログファイルでは、アンインストール中に構成ファイルと証明書は削除されません。それらを削除したい場合は、同じフォルダでショートカット **UninstallAll.lnk** を実行します。

優れた可視性または適用 AIX エージェントの削除

手順

コマンド **'installp -u tet-sensor'** を実行します。

エージェントのアンインストールイベントがクラスタに通知され、[ソフトウェアエージェント（Software Agent）] ページでエージェントがアンインストール済みとしてマークされます。

[ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを手動で削除します。また、ユーザーは、エージェント設定プロファイルから [クリーンアップ期間（cleanup period）] をオンにして、エージェントの自動クリーンアップまたは削除を有効にすることもできます。

（注）
- デフォルトでは、[クリーンアップ期間（cleanup period）] はオフになっています。
- 優れた可視性エージェントは、System Resource Controller によって tet-sensor として制御されます。起動、停止、再起動、削除が可能です。このサービスは、inittab を tet-sen-engine として使用して永続化されます。
- 適用エージェントは、System Resource Controller によって tet-enforcer として制御されます。起動、停止、再起動、削除が可能です。このサービスは、inittab を tet-enf-engine として使用して永続化されます。
- エージェントの操作中に、一部のカーネルモジュールがカーネルによって自動的にロードされる可能性があります。たとえば、AIX で適用が有効になっている場合、ipfilter モジュールがロードされます。エージェントには、カーネルによってロードされたモジュールのリストがありません。したがって、エージェントのアンインストール中に、カーネルモジュールをアンロードできない可能性があります。
- 適用エージェントがシステムファイアウォールにポリシーを適用した場合、エージェントをアンインストールすると、適用されたポリシーがクリアされ、システムファイアウォールが開きます。

ユニバーサル Linux エージェントの削除

手順

ステップ 1 アンインストールスクリプト **‘/usr/local/tet-light/uninstall.sh‘** を実行します。
ステップ 2 [ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを削除します。

ユニバーサル Windows エージェントの削除

手順

ステップ 1 アンインストールスクリプト **‘C:\Program Files\Cisco Tetration\Lightweight Sensor\uninstall.cmd‘** を実行します。
ステップ 2 [ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを削除します。

適用 Kubernetes または OpenShift エージェントの削除

手順

ステップ 1 元のインストーラスクリプトを見つけるか、Secure Workload UI から新しいスクリプトをダウンロードします。

ステップ 2 アンインストールオプション **install.sh –uninstall** を実行します。インストール時と同じ考慮事項が適用されます。

- Linux x86_64 アーキテクチャでのみサポートされます。
- ~/.kube/config に含まれている管理者ユーザーログイン情報を使用するか、または–kubeconfig オプションを使用して kubectl 管理者ログイン情報ファイルを指定します。

ステップ 3 [ソフトウェアエージェント（Software Agent）] ページの UI からすべての Kubernetes ノードのエージェントを削除します。

優れた可視性 Solaris エージェントの削除

手順

ステップ 1 コマンド `pkg uninstall tet-sensor` を実行します。

ステップ 2 [ソフトウェアエージェント（Software Agent）] ページでエージェントを削除します。

ワークロードエージェントにより収集されエクスポートされるデータ

このセクションでは、ソフトウェアエージェントの主要コンポーネント、バックエンドサービスへの登録方法、分析目的で収集されてクラスタにエクスポートされるデータについて説明します。

登録

エージェントがシステムに正常にインストールされたら、有効な一意の識別子を取得するために、エージェントをバックエンドサービスに登録する必要があります。次の情報が登録要求で送信されます。

- ホスト名

- BIOS-UUID
- プラットフォーム情報（CentOS-6.5 など）
- 自己生成クライアント証明書（openssl コマンドで生成）
- エージェントタイプ（可視性または適用）

エージェントがサーバーから有効な ID を取得できなかった場合、取得するまで再試行を続けます。エージェントの登録は非常に重要です。登録されていない場合、その後に続く他のサービス（コレクタなど）との通信がすべて拒否されます。

エージェントのアップグレード

エージェントは定期的（約 30 分）にバックエンドサービスにメッセージを送信して、現在のバージョンを報告します。バックエンドサービスは、エージェントの ID とその現在のバージョンを使用して、新しいソフトウェアパッケージがエージェントで使用できるかどうかを判断します。次の情報が送信されます。

- エージェントの ID （登録成功後に取得）
- 現在のエージェントのバージョン

構成サーバー

エージェントは、構成された構成サーバーに次の情報をエクスポートします。

- Hostname
- エージェントの ID（登録成功後に取得）
- インターフェイスのリスト。各インターフェイスに含まれる内容：
  1. インターフェイスの名前
  2. IP ファミリ（IPv4 または IPv6）
  3. IP アドレス
  4. ネットマスク
  5. MAC アドレス
  6. インターフェイスのインデックス

インターフェイスプロパティが変更されると（既存のインターフェイスの IP アドレスの変更、新しいインターフェイスの起動など）、このリストが更新され、構成サーバーに報告されます。

ネットワークフロー

ネットワークフロー情報は、システムを流れるすべてのパケットを要約したものです。フロー情報をキャプチャするには、詳細と会話の 2 つのモードがあります。デフォルトでは、詳細モードのキャプチャが使用されます。キャプチャされたフローはコレクタにエクスポートされ、エクスポートされる情報には次のものが含まれます。

- フロー識別子：ネットワークフローを一意に識別します。これには、IPプロトコル、送信元と宛先の IP、レイヤー 4 ポートなどの一般情報が含まれます。
- IP 情報：TTL、IP フラグ、パケット ID、IP オプション、フラグメンテーションフラグなど、IP ヘッダーに表示される情報が含まれます。
- TCP 情報：シーケンス番号、ACK 番号、TCP オプション、Rcvd ウィンドウサイズなど、TCP ヘッダーに表示される情報が含まれます。
- フロー情報：フローの統計情報（合計パケット数、合計バイト数、TCPフラグ統計情報、パケット長統計情報、ソケット統計情報など）、フローが観察されたインターフェイスインデックス、フローの開始時間と終了時間

カンバセーションモードでは、エージェントは他のコネクションレス型フローとともに、本質的に双方向のTCPフローのみをエクスポートします。カンバセーションモードは、Windows、AIX、および Linux プラットフォームでサポートされています。カンバセーションモードの詳細については、「Conversation Mode」を参照してください。

（注） K8s 環境では、ポッド/ホストフローの関連付けはカンバセーションモードでは実行されません。

どちらのモードでも、エージェントは次のフローをエクスポートしないことに注意してください。

- ARP/RARP 会話
- エージェントからコレクタへのフロー

マシン情報

マシン情報には、ホストで実行されているすべてのプロセスが記述されています。さらに、プロセスに関連付けられたネットワーク情報と、プロセスの起動に使用されるコマンドが含まれています。マシン情報は毎分エクスポートされ、次の情報が含まれます。

- [プロセス ID（Process ID）]
- [ユーザーID（User ID）]：プロセスの所有者
- [親プロセス ID（Parent Process ID）]
- プロセスの起動に使用されるコマンド文字列

- [ソケット情報（Socket information）]：プロトコル（UDP または TCP など）、アドレスタイプ（IPv4 または IPv6）、送信元および宛先 IP、送信元および宛先ポート、TCP 状態、プロセスの開始時刻と終了時刻、バイナリを処理するパス
- [フォレンジック情報（Forensic information）]：詳細については、「互換性（698 ページ）」のセクションを参照してください。

エージェント統計情報

エージェントは、システムの統計や独自の統計を含む、次のようなさまざまな統計を追跡します。

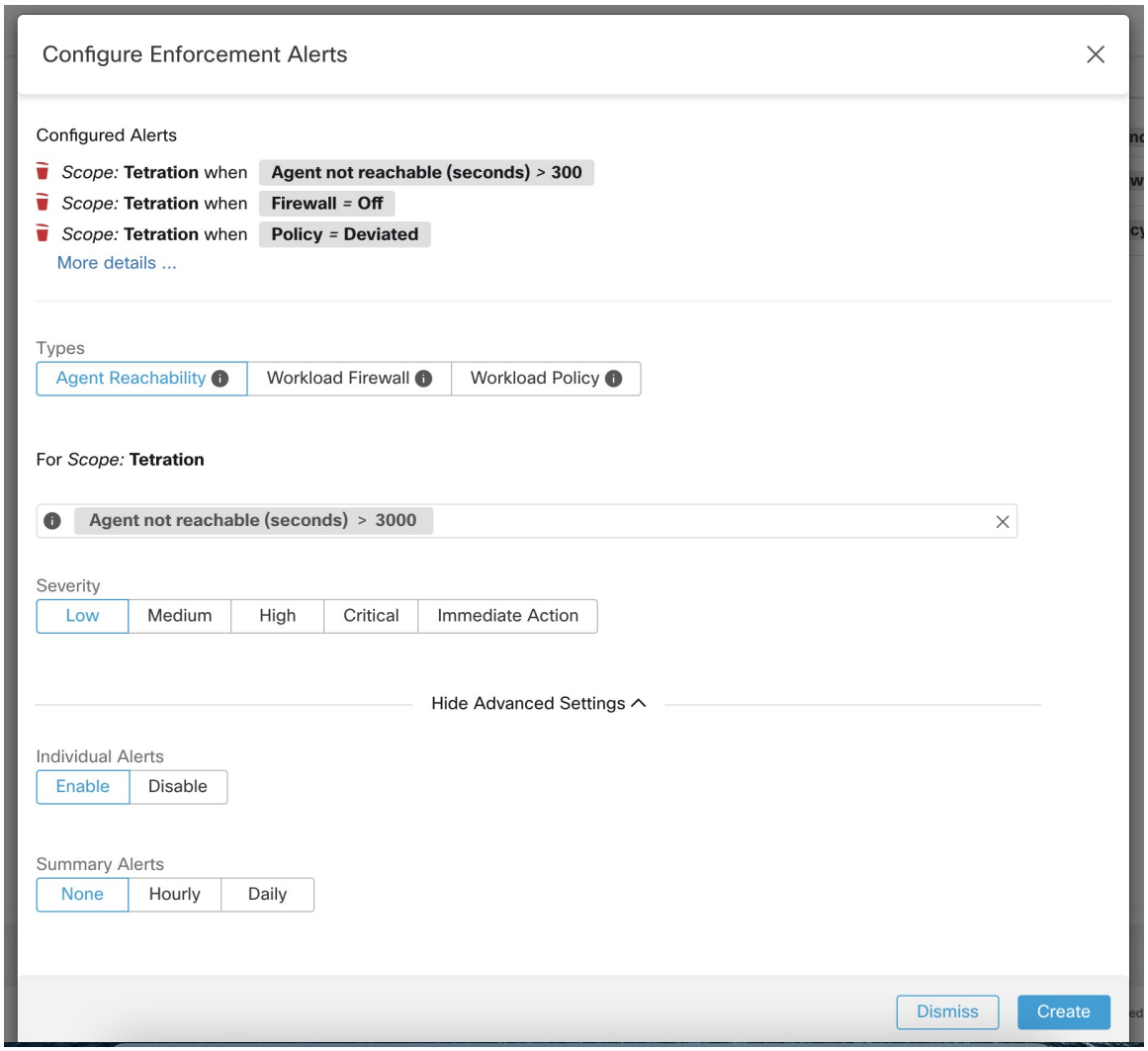
- エージェントの開始時間と稼働時間
- ユーザーモードとカーネルモードでのエージェントの実行時間
- ドロップされた受信データパケットの数
- 成功および失敗した SSL 接続の数
- 総フローパケット数とバイト数
- コレクタにエクスポートされたフローとパケットの合計
- エージェントのメモリおよび CPU の使用状況

適用アラート

適用アラートには次の 3 つのタイプがあります。

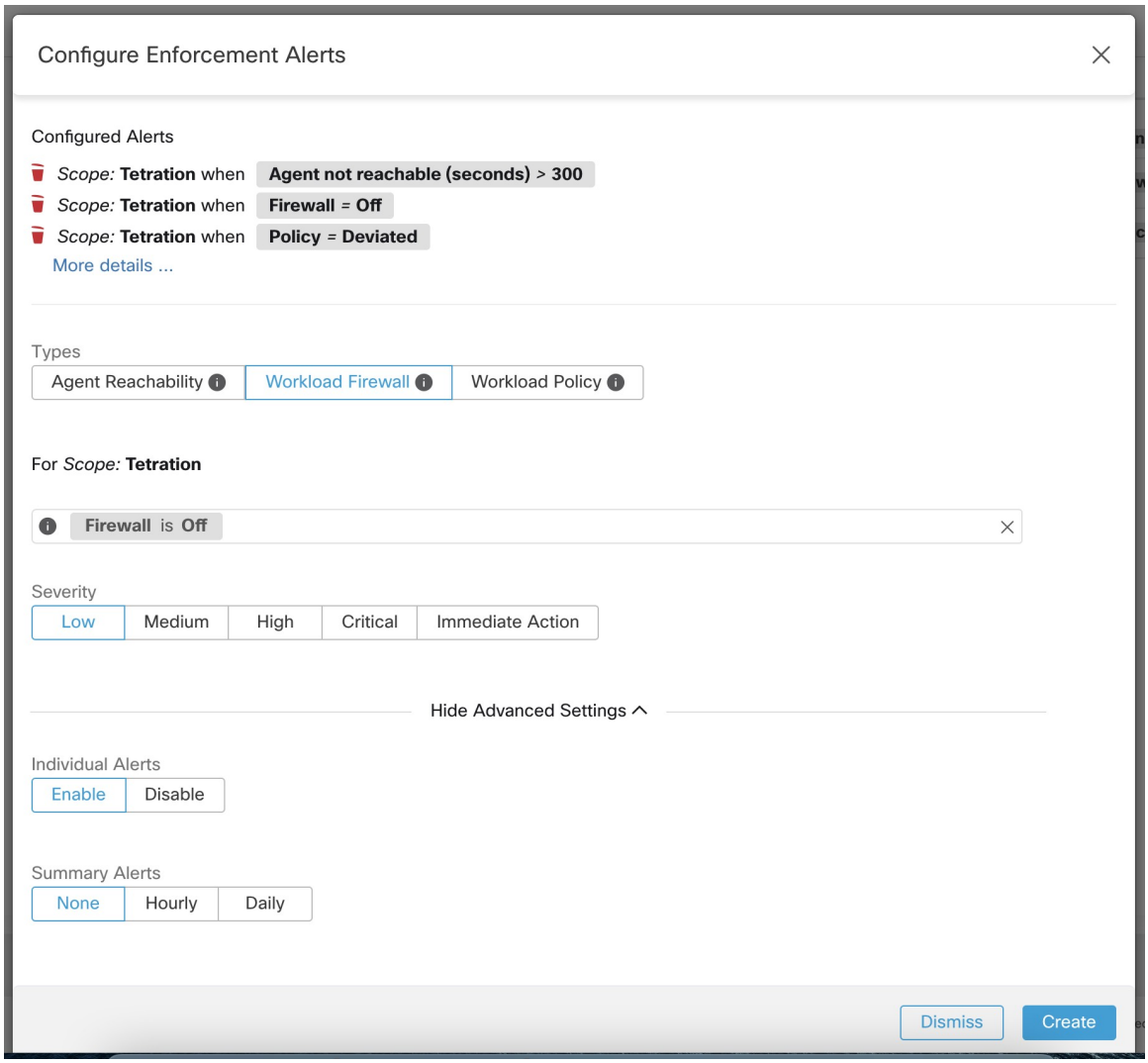
- エージェントの到達可能性

  このアラートは、エージェントに到達できない場合を検知します。このアラートは、エージェントが設定された秒数を超えて Secure Workload クラスタと通信しなかった場合にトリガーされます。



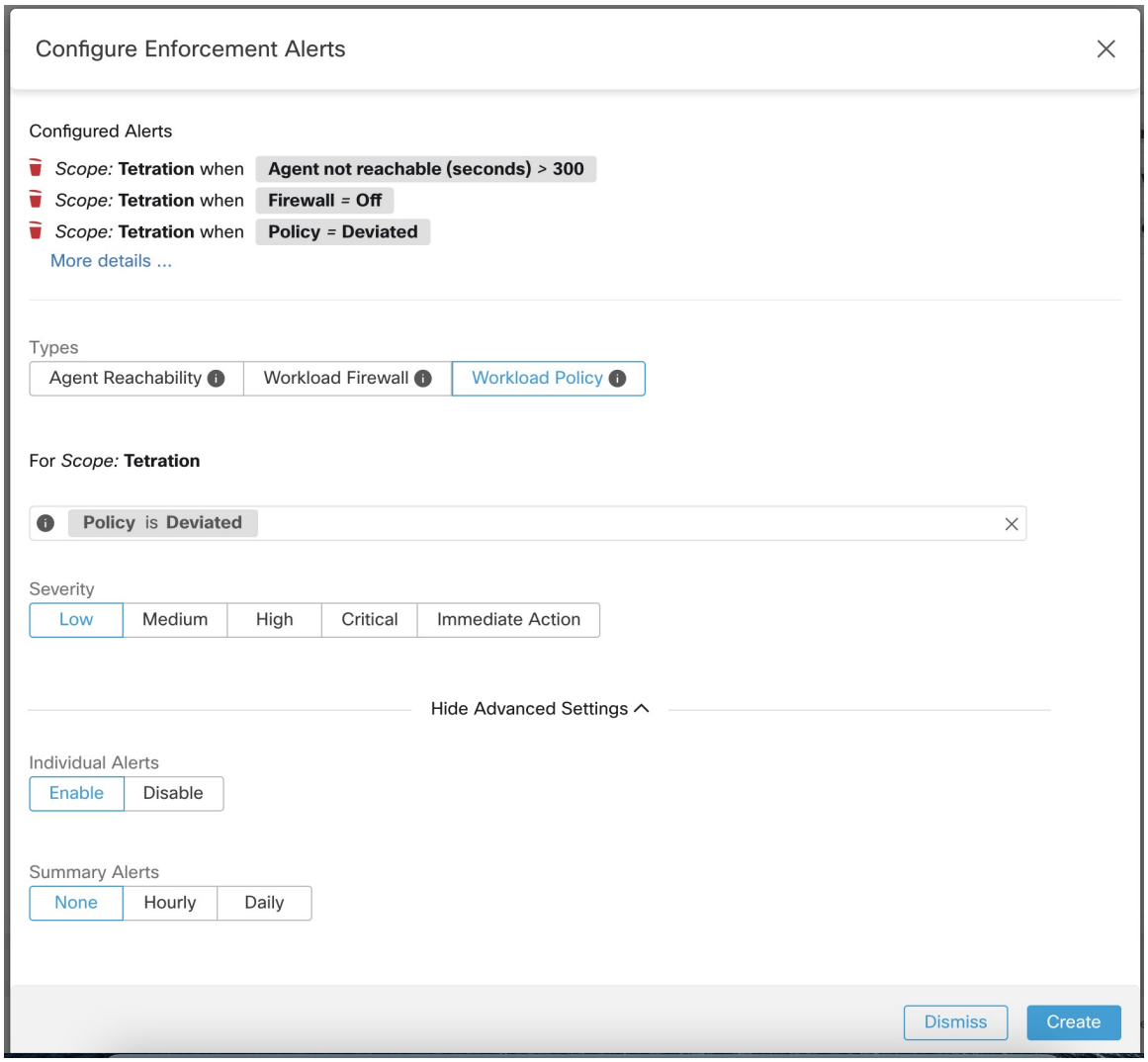
- ワークロード ファイアウォール

  このアラートは、ワークロードで適用が設定されているものの、ワークロード ファイアウォールがオフであることが検知された場合にトリガーされます。この状態では、Secure Workload エージェントがトラフィックポリシーを適用できなくなるためです。



- ワークロードポリシー

  このアラートは、ワークロードファイアウォールのルールが、このワークロードに適用可能な Secure Workload ポリシー（ワークロードの「具体的なポリシー」）と異なる場合にトリガーされます。



アラートの[重大度（Severity）]およびその他のタイプごとの設定パラメータを設定できます。

適用アラートを設定するには、アラート （779 ページ）を参照してください。

図 *27* : 適用アラートの設定

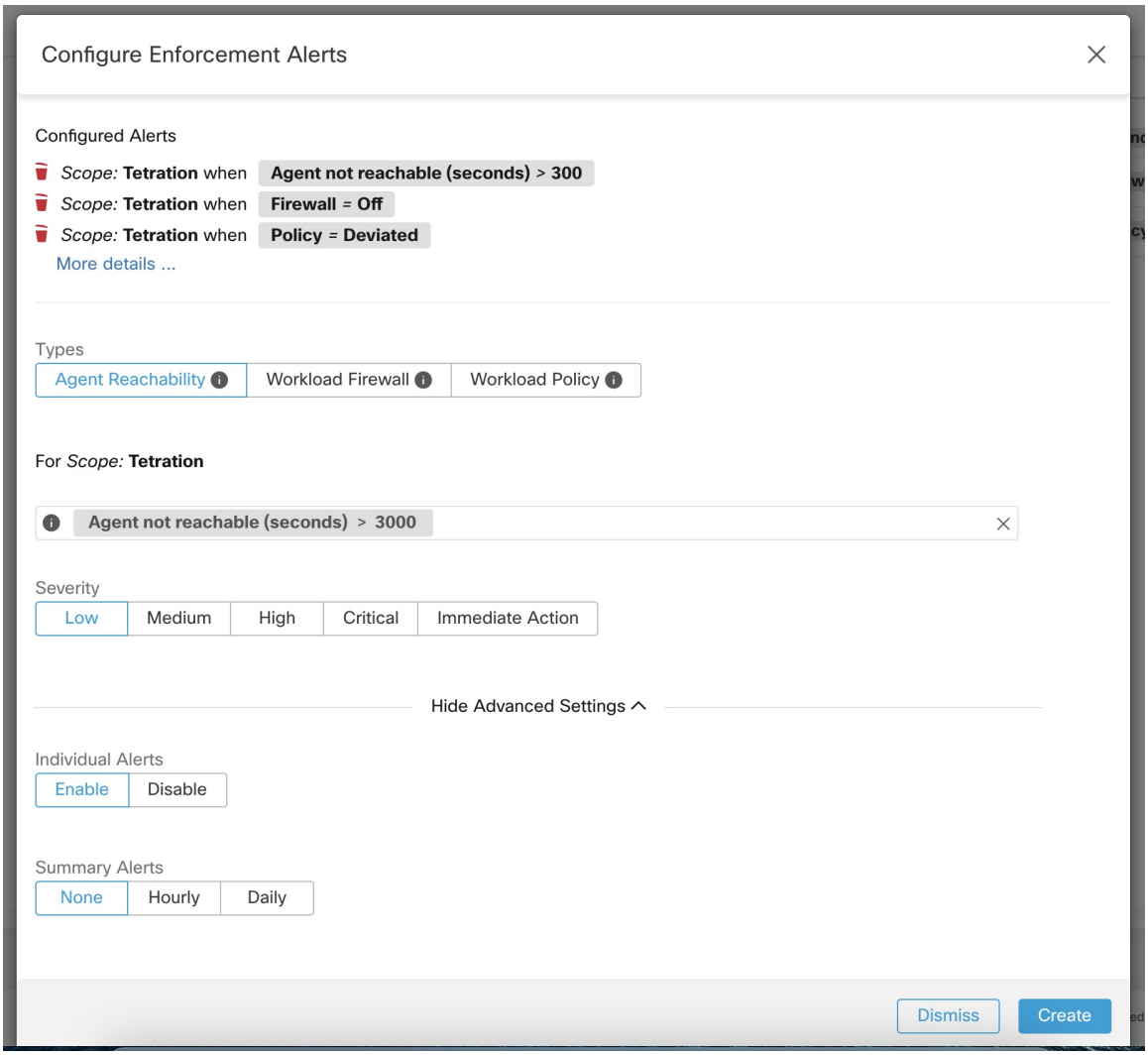
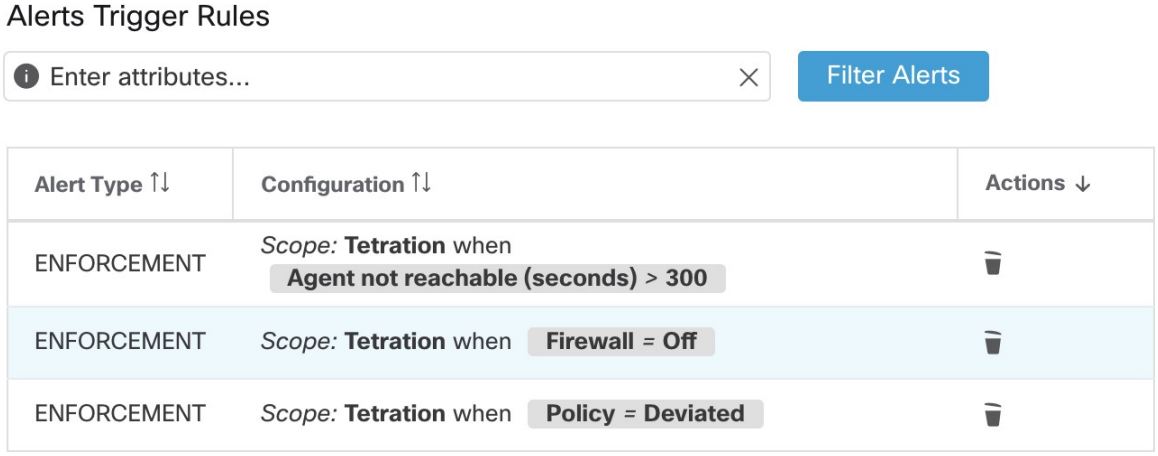


図 *28* : アラート設定ページでの設定済み適用アラートの表示



適用 UI アラートの詳細

図 *29* : 適用アラートの詳細

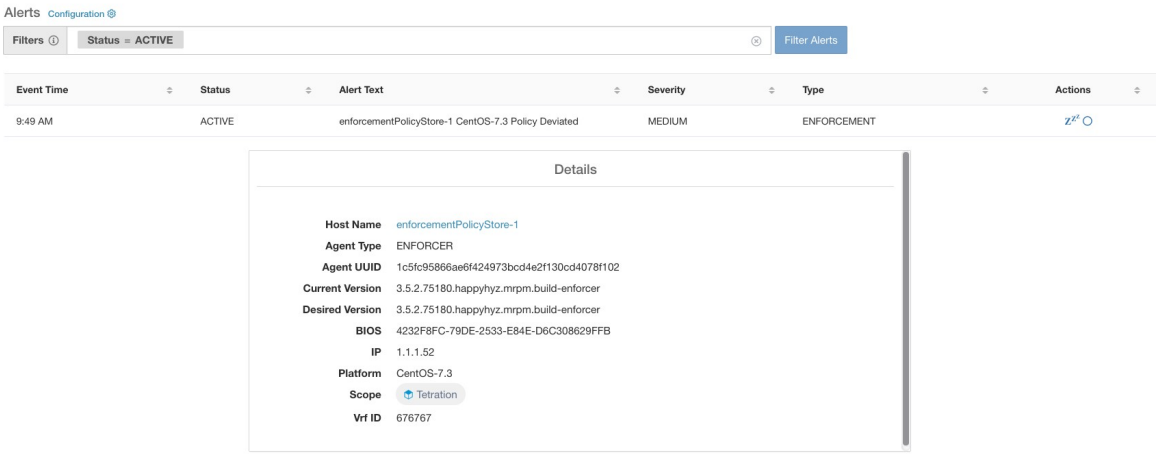
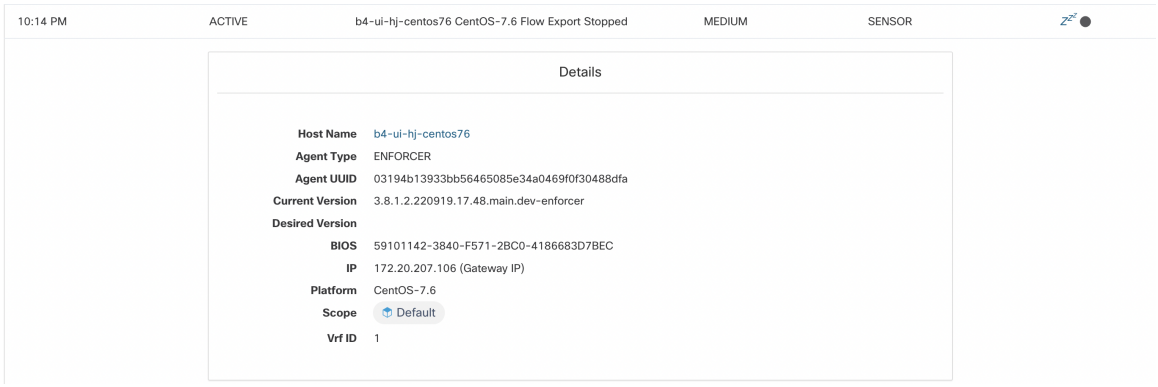


図 *30* : ホストでプロキシが有効になっている場合の適用アラートの詳細



適用アラートの詳細

一般的なアラート構造とフィールドに関する情報については、「一般的なアラート構造」を参照してください。alert_details フィールドは構造化されており、適用アラートの次のサブフィールドが含まれています。

フィールド	アラートタイプ	フォーマット	説明
AgentType	*all*	string	インストールタイプに応じて「ENFORCER」または「SENSOR」
HostName	*all*	string	エージェントが展開されているホスト名

フィールド	アラートタイプ	フォーマット	説明
IP	*all*	string	ノードまたはゲートウェイの IP アドレス
Bios	*all*	string	ノードの BIOS UUID
プラットフォーム	*all*	string	ノードのプラットフォームまたは OS 情報
CurrentVersion	*all*	string	ノード上のエージェントのソフトウェアバージョン
DesiredVersion	*all*	string	エージェントに必要なソフトウェアバージョン
LastConfigFetchAt	*all*	integer	エージェントが最後に HTTPS リクエストを送信したときの UNIX タイムスタンプ

適用アラートの alert_details の例

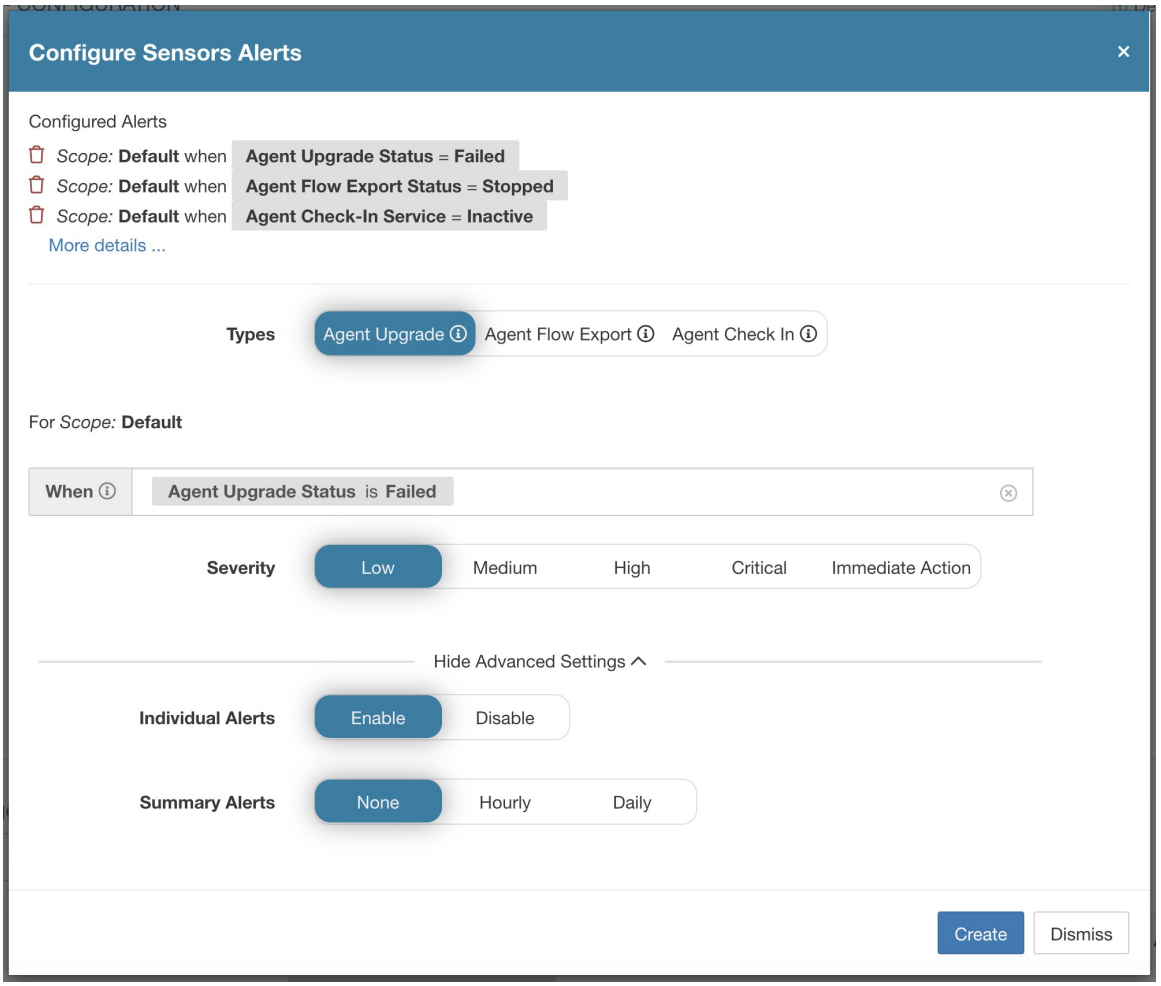
```
{
  "AgentType":"ENFORCER",
  "Bios":"72EF1142-03A2-03BC-C2F8-F600567BA320",
  "CurrentVersion":"3.5.1.1.mrpm.build.win64-enforcer",
  "DesiredVersion":"",
  "HostName":"win2k12-production-db",
  "IP":"172.26.231.193",
  "Platform":"MSServer2012R2Standard"
}
```

センサーアラート

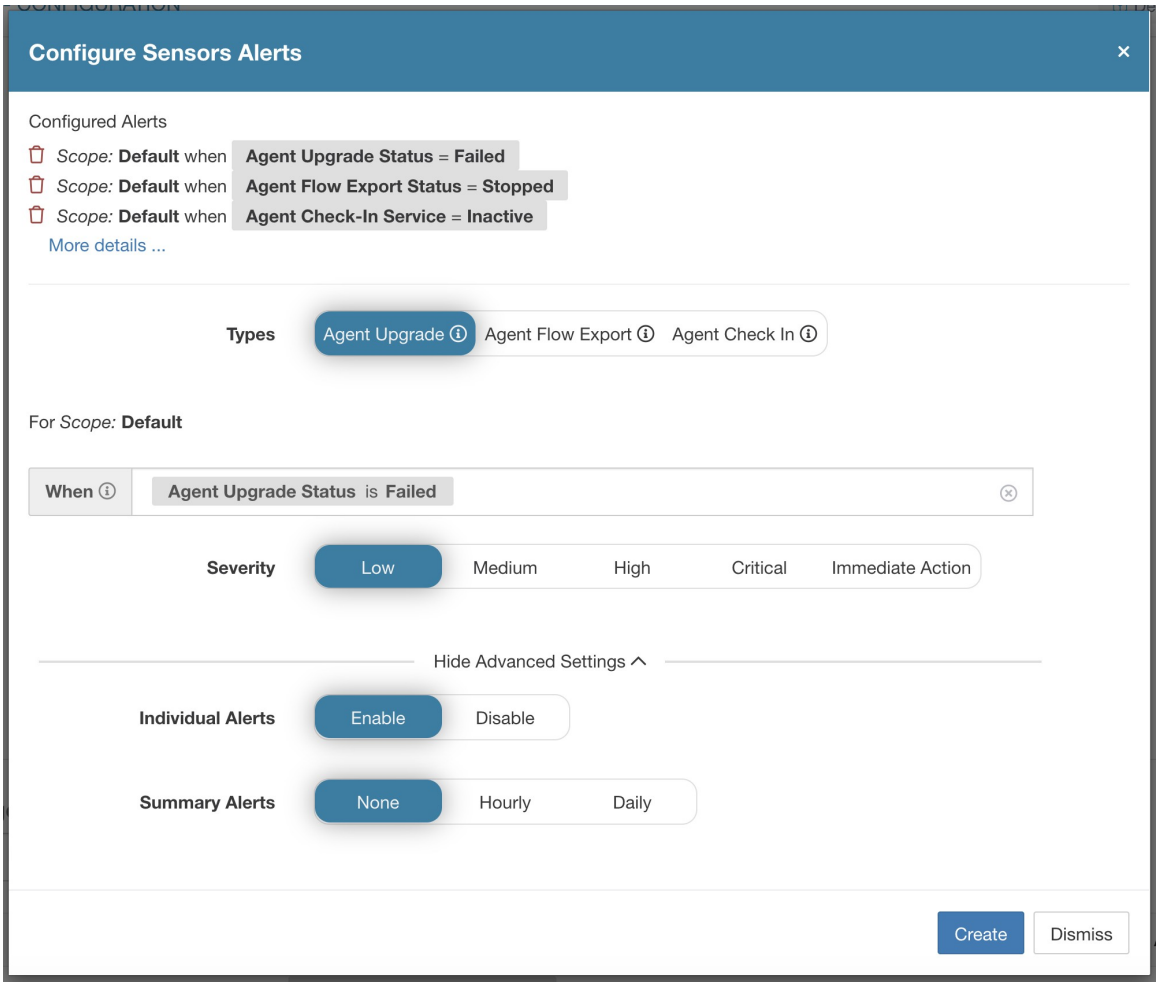
（注） リリース 3.5 以降、センサーアラートはアラート設定モデルを使用して設定できます。

センサーアラートはアラート設定モデルを使用して設定できます。モデルに関する一般的な情報については、「アラート設定モーダル」を参照してください。

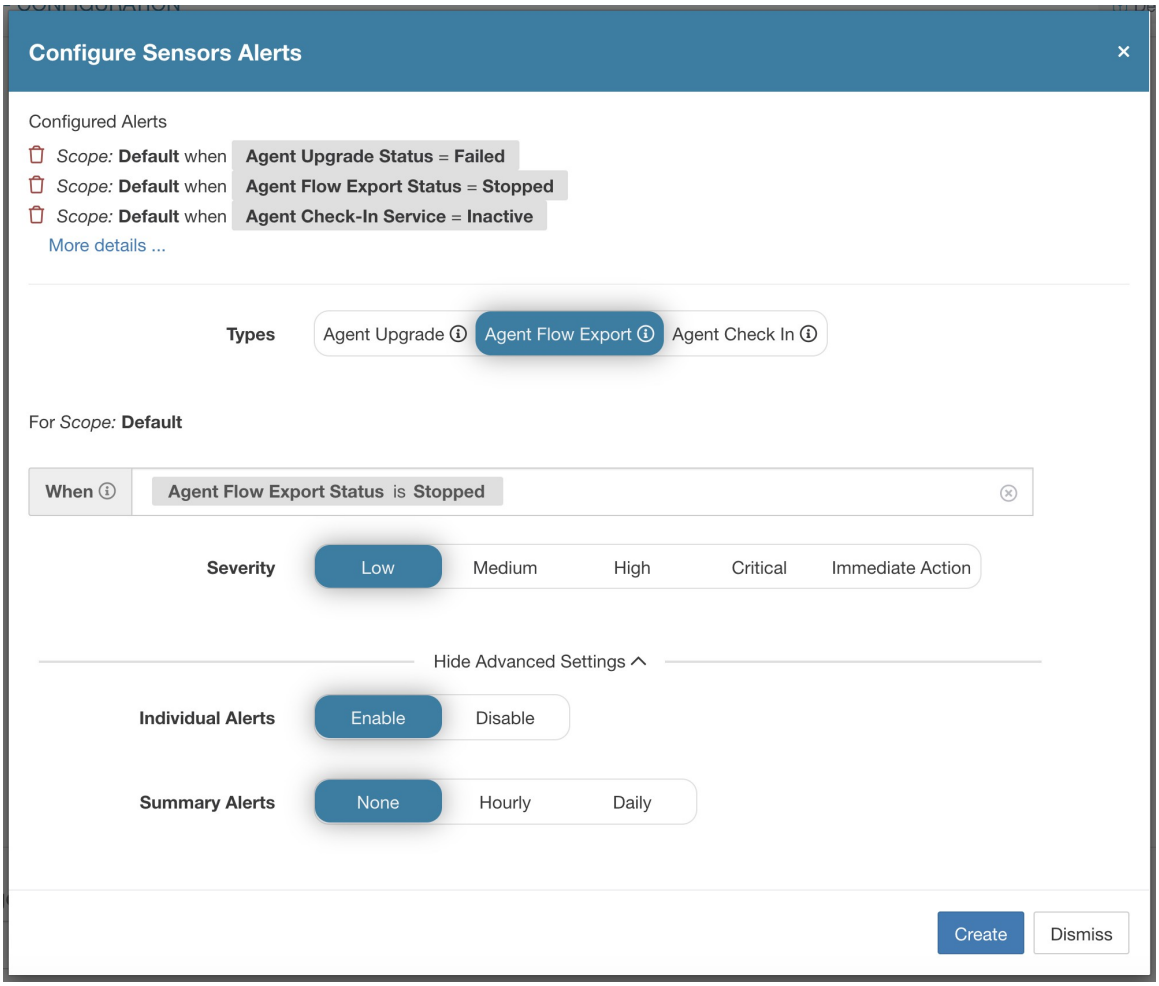
図 *31* : センサーアラートの設定



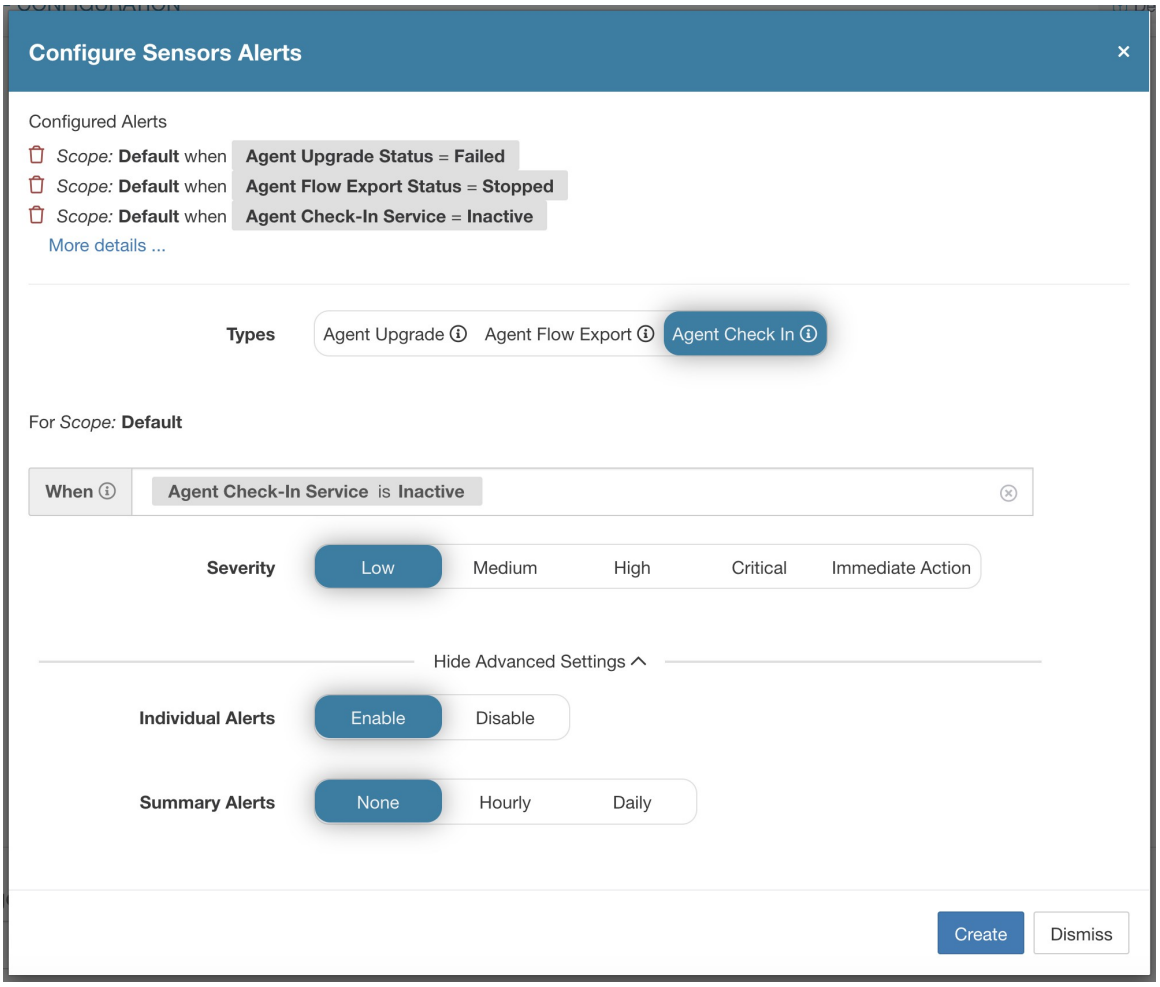
センサーアラート設定には、3 つの異なるタイプのアラートを設定する機能があり、ユーザーはアラートの重大度やその他のタイプごとの設定パラメータを設定できます。



エージェントがアップグレードに失敗したときに報告するようにセンサーアラートを設定します。このアラートは、エージェントによる目的のバージョンへのアップグレードに失敗した場合にトリガーされます。

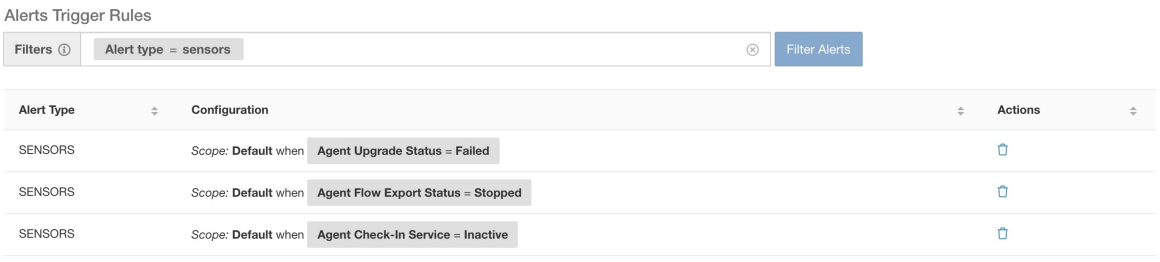


エージェントフローのエクスポートが停止したことを検出するようにセンサーアラートを設定します。このアラートは、エージェントとクラスタ間の接続がどこかでブロックされ、フローやその他のシステム情報の送信または配信が妨げられている場合にトリガーされます。



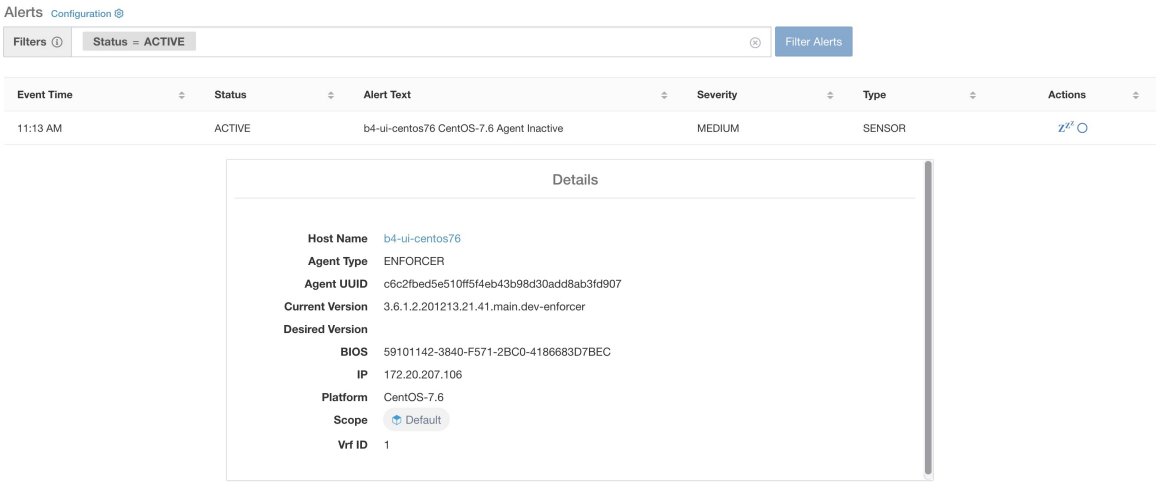
エージェントのチェックインのタイムアウトを検出するようにセンサーアラートを設定します。このアラートは、クラスタがエージェントからのチェックイン要求を 90 分以上受信していない場合にトリガーされます。

図 *32* : アラート設定ページでの設定済みセンサーアラートの表示



センサー UI アラートの詳細

図 *33* : センサーアラートの詳細



センサーアラートの詳細

一般的なアラート構造とフィールドに関する情報については、「一般的なアラート構造」を参照してください。alert_details フィールドは構造化されており、センサーアラートの次のサブフィールドが含まれています。

フィールド	アラートタイプ	フォーマット	説明
AgentType	*all*	string	インストールタイプに応じて「ENFORCER」または「SENSOR」
HostName	*all*	string	エージェントが展開されているホスト名
IP	*all*	string	ノードまたはゲートウェイの IP アドレス
Bios	*all*	string	ノードの BIOS UUID
プラットフォーム	*all*	string	ノードのプラットフォームまたは OS 情報
CurrentVersion	*all*	string	ノード上のエージェントのソフトウェアバージョン

フィールド	アラートタイプ	フォーマット	説明
DesiredVersion	*all*	string	エージェントに必要なソフトウェアバージョン
LastConfigFetchAt	*all*	integer	エージェントが最後にHTTPSリクエストを送信したときのUNIXタイムスタンプ

センサーアラートの alert_details の例

```
{
  "AgentType":"SENSOR",
  "Bios":"72EF1142-03A2-03BC-C2F8-F600567BA320",
  "CurrentVersion":"3.5.1.1.mrpm.build.win64-sensor",
  "DesiredVersion":"",
  "HostName":"win2k12-production-db",
  "IP":"172.26.231.193",
  "Platform":"MSServer2012R2Standard"
}
```

ソフトウェアエージェントのトラブルシューティング

ここでは、ソフトウェアエージェントの展開中や運用中にお客様が直面する可能性のあるいくつかの潜在的な問題について取り上げます。また、問題のトラブルシューティングに使用できる方法、およびお客様が適用できるいくつかの解決策を記載します。

一般

ログファイル：ログファイルは通常、<install-location>/logs または <install-location>/log フォルダに保存されます。これらのログファイルは、Secure Workload サービスによって監視およびローテーションされます。

エージェントの展開

Linux

Q：コマンド「rpm -Uvh tet-sensor-1.101.2-1.el6-dev.x86_64.rpm」を実行すると、エージェントのインストールに失敗し、次のエラーがスローされました。

```
error: can't create transaction lock on /var/lib/rpm/.rpm.lock (Permission denied).
```

A：エージェントをインストールするための適切な権限がないようです。rootに切り替えるか、sudo を使用してエージェントをインストールしてください。

Q：「sudo rpm -Uvh tet-sensor-1.0.0-121.1b1bb546.el6-dev.x86_64.rpm」を実行すると、次のエラーが発生しました。

```
   Preparing...                ########################################### [100%]
   which: no lsb_release in (/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin)
   error: %pre(tet-sensor-site-1.0.0-121.1b1bb546.x86_64) scriptlet failed, exit status
 1
   error:   install: %pre scriptlet failed (2), skipping tet-sensor-site-1.0.0-121.1b1bb546
```

A：システムはエージェントをインストールするための要件を満たしていません。今回のケースでは、lsb_release ツールはインストールされていません。詳細については sw_agents_deployment-label セクションを参照し、必要な依存関係をインストールしてください。

Q：「sudo rpm -Uvh tet-sensor-1.0.0-121.1b1bb546.el6-dev.x86_64.rpm」を実行すると、次のエラーが発生しました。

```
  Unsupported OS openSUSE project
  error: %pre(tet-sensor-1.101.1-1.x86_64) scriptlet failed, exit status 1
  error: tet-sensor-1.101.1-1.x86_64: install failed
  warning: %post(tet-sensor-site-1.101.1-1.x86_64) scriptlet failed, exit status 1
```

A：お使いの OS では、ソフトウェアエージェントの実行がまだサポートされていません（今回のケースでは、「openSUSE project」はサポートされていないプラットフォームです）。詳細については sw_agents_deployment-label セクションを参照してください。

Q：すべての依存関係がインストールされており、適切な権限でインストールを実行しました。インストールはうまくいき、エラーはスローされませんでした。エージェントのインストールが本当に成功したことを確認するにはどうすればよいですか。

A：エージェントのインストール後に、このコマンドを実行して確認できます。

```
$ ps -ef | grep -e tet-sensor -e tet-engine
root 12655 1 0 08:26 ? 00:00:00 tet-engine
root 12659 12655 0 08:26 ? 00:00:00 tet-engine check_conf
root 12660 12655 0 08:26 ? 00:00:00 tet-sensor -f sensor.conf
```

3 つのエントリが表示されます。2 つは tet-engine プロセス用で、1 つは tet-sensor プロセス用です。それらのエントリが実行されていない場合は、ディレクトリ /usr/local/tet が存在しているかどうかを確認してください。存在していない場合は、インストールが失敗している可能性があります。

```
$ ps -ef | grep -e csw-agent -e tet-
root     14158     1  0 Apr03 ?        00:00:00 csw-agent
root     14160 14158  0 Apr03 ?        00:00:00 csw-agent watch_files
root     14161 14158  0 Apr03 ?        00:00:03 csw-agent check_conf
root     14162 14158  0 Apr03 ?        00:01:03 tet-sensor -f conf/.sensor_config
root     14163 14158  0 Apr03 ?        00:02:38 tet-main --sensoridfile=./sensor_id
root     14164 14158  0 Apr03 ?        00:00:22 tet-enforcer --logtostderr
tet-sen+ 14173 14164  0 Apr03 ?        00:00:21 tet-enforcer --logtostderr
tet-sen+ 14192 14162  0 Apr03 ?        00:07:23 tet-sensor -f conf/.sensor_config
```

csw-agent の 3 つのエントリと、tet-sensor の少なくとも 2 つのエントリが表示されている必要があります。サービスが実行されていない場合は、次のディレクトリが使用可能であることを確認します。使用可能でない場合、インストールは失敗します。

- `/usr/local/tet`（ほとんどの Linux ディストリビューションの場合）
- `/opt/cisco/tetration`（AIX、Ubuntu の場合）
- `/opt/cisco/secure-workload`（Solaris、Debian の場合）

Windows

Q：PowerShell エージェントインストーラ スクリプトを実行すると、次のいずれかのエラーが表示されます。

1. 基礎となる接続がクローズしました。受信時に予期せぬエラーが発生しました。
2. 共通のアルゴリズムがないため、クライアントとサーバーが通信できません

A：ホストとサーバーで設定されている SSL/TLS プロトコルが不一致である可能性が高いです。次のコマンドを使用して、SSL/TLS バージョンを確認できます。

```
[Net.ServicePointManager]::SecurityProtocol
```

サーバーと一致するように SSL/TLS を設定するには、次のコマンドを使用できます（これは永続的な変更ではなく、現在の PowerShell セッションでの一時的な変更であることに注意してください）。

```
[Net.ServicePointManager]::SecurityProtocol =
[System.Net.SecurityProtocolType]'Ssl3,Tls,Tls11,Tls12'
```

Q：ダウンロードしたバンドルから MSI インストーラを実行すると、次のエラーが表示されます。

```
This installation package could not be opened. Verify that the package exists and that
you can access it, or contact the application vendor to verify that this is a valid
Windows Installer package.
```

A：*C:\Windows\Installer* パスが存在することを確認してください。コマンドラインから MSI インストーラを実行する場合は、msi ファイルを指定するときに相対パスを含めないようにしてください。正しい構文の例を示します。

```
msiexec /i "TetrationAgentInstaller.msi" /l*v "msi_install.log" /norestart
```

Q：基盤となる NIC が Nutanix VirtIO ネットワークドライバの場合、Windows Sensor ソフトウェアのアップグレードに失敗するようです。

A：Npcap 0.9990 と、Nutanix VirtIO ネットワークドライバ 1.1.3 以前のバージョンの間には、非互換性の問題があります。また、Receive Segment Coalescing が有効になっています。

この問題を解決するには、Nutanix VirtIO ネットワークドライバをバージョン 1.1.3 以降にアップグレードします。

Q：Windows Sensor をインストールしました。そのセンサーが登録されていないようであり、sensor_id ファイルには uuid-invalid-platform が含まれています。

A：Windows の PATH 変数に system32 がない可能性があります。system32 が PATH にあるかどうかを確認してください。ない場合は、次のコマンドを実行します。

```
set PATH=%PATH%;C:\Windows\System32\
```

Kubernetes

Kubernetes Daemonset のインストール中にインストーラスクリプトが失敗する場合、多くの理由が考えられます。

Q：ノードから到達可能なイメージを提供する Docker レジストリはありますか。

A：Cisco Secure Workload クラスタからイメージをプルするクラスタに関するダイレクトまたは HTTPS プロキシの問題をデバッグします。

Q：コンテナランタイムは SSL/TLS の安全でないエラーを報告していますか。

A：コンテナランタイムの適切な場所にあるすべての Kubernetes ノードに、Secure Workload HTTPS CA 証明書がインストールされていることを確認します。

Q：Docker レジストリ認証とイメージダウンロードの許可が失敗しましたか。

A：各ノードから、Helm チャートによって作成されたシークレットからの Docker プルシークレットを使用して、Daemonset 仕様のレジストリ URL から Docker が手動でイメージをプルするようにしてみてください。手動でのイメージプルも失敗した場合は、問題をさらにデバッグするため、Secure Workload クラスタの registryauth サービスからログをプルする必要があります。

Q：Kubernetes クラスタは Secure Workload アプライアンス内で正常にホストされていますか。

A：クラスタの [サービスステータス（service status）] ページをチェックして、関連するすべてのサービスが正常であることを確認します。[探索（explore）] ページから dstool スナップショットを実行し、生成されたログを取得します。

Q：Docker イメージビルダデーモンは実行されていますか。

A：dstool ログから、ビルドデーモンが実行されていることを確認します。

Q：Docker イメージをビルドするジョブは失敗しますか。

A：dstool ログから、イメージがビルドされていないことを確認します。Docker ビルドポッドログを使用して、ビルドキットを構築中のエラーをデバッグできます。適用コーディネータのログを使用して、ビルドの失敗をさらにデバッグすることも可能です。

Q：Helm チャートを作成するジョブは失敗しますか。

A：dstool ログから、Helm チャートが構築されていないことを確認します。適用コーディネータのログには Helm 構築ジョブの出力が含まれます。Helm チャートの構築ジョブの失敗の正確な理由をデバッグするために使用できます。

Q：インストール bash スクリプトが壊れていましたか。

A：インストールbashスクリプトのダウンロードを再試行します。bashスクリプトには、追加のバイナリデータが含まれています。bashスクリプトをテキストエディタで編集したり、テキストファイルとして保存したりすると、バイナリデータの特殊文字がテキストエディタによって破損または変更される可能性があります。

Q：Kubernetes クラスタ設定：亜種とフレーバーが多すぎます。クラシック K8 をサポートしています。

A：お客様が Kubernetes の亜種を実行している場合、展開のさまざまな段階で多くの障害モードが発生する可能性があります。失敗の段階を分類します：kubectl コマンド実行の失敗、helm コマンド実行の失敗、ポッドイメージのダウンロードの失敗、ポッドの特権モードオプションの拒否、ポッドイメージの信頼コンテンツ署名の失敗、ポッドイメージのセキュリティスキャンの失敗、ポッドバイナリ実行の失敗（アーキテクチャの不一致）、ポッドを実行しても Secure Workload サービスの開始が失敗、Secure Workload サービスが開始しても異常な動作環境が原因でランタイムエラーが発生。

Q：Kubernetes RBAC 資格情報が失敗していますか。

A：特権 DaemonSet を実行するには、K8s クラスタに対する管理者権限が必要です。kubectl 設定ファイルに、ターゲットクラスタおよびそのクラスタの管理者と同等のユーザーを指すデフォルトのコンテキストがあることを確認します。

Q：BusyBox イメージは、すべてのクラスタノードから利用可能またはダウンロード可能ですか。

A：接続の問題を修正し、BusyBoxイメージがダウンロードできることを手動でテストします。ポッド仕様で使用されているBusyBoxの正確なバージョンは、すべてのクラスタノードで使用可能（事前シード済み）またはダウンロード可能である必要があります。

Q：インストール中に、API サーバーおよび etcd エラーまたは通常のタイムアウトが発生しましたか。

A：Kubernetesクラスタ内のすべてのノードでDaemonSetポッドがインスタンス化されるため、クラスタの CPU/ディスク/ネットワークの負荷が突然スパイクする可能性があります。この問題は、お客様固有のインストールの詳細に大きく依存します。過負荷が原因で、インストールプロセス（すべてのノードでプルされてディスクに書き込まれるイメージ）に時間がかかりすぎたり、Kubernetes API サーバー、または Secure Workload Docker レジストリエンドポイントが過負荷になったり、プロキシサーバー（設定されている場合）が一時的に過負荷になったりする可能性があります。すべてのノードでイメージのプルが完了し、Kubernetes クラスタノードの CPU/ディスク/ネットワークの負荷が軽減されるのを少し待ってから、インストールスクリプトを再試行します。API サーバーと Kubernetes コントロールプレーンからの etcd エラーは、Kubernetes コントロールプレーンノードがアンダープロビジョニングであるか、アクティビティの突然のスパイクの影響を受けている可能性があることを示しています。

Q：Secure Workload エージェントの操作でランタイムの問題が発生していますか。

A：ポッドが正しく展開され、エージェントの実行が開始されているが、ランタイムの問題が発生している場合は、Linux エージェントのトラブルシューティング セクションを参照してください。Kubernetesの展開が正常にインストールされ、ポッドが開始された後のトラブルシューティング手順は同じです。

異常タイプ

これらは、Secure Workloadエージェントの使用および管理時にワークフローで発生する最も一般的な問題です。

非アクティブなエージェント

エージェントによるクラスタサービスへのチェックが停止しています。この現象は次のような原因で発生する可能性があります。

- ホストがダウンしている可能性がある
- ネットワーク接続が壊れているか、ファイアウォールルールによってブロックされている
- エージェントサービスが停止している

すべてのプラットフォーム

- ホストがアクティブで正常であることを確認します
- エージェントサービスが稼働状態であることを確認します
- クラスタへのネットワーク接続が機能していることを確認します

アップグレードの失敗

エージェントのアップグレードに失敗しました。これは、次のようないくつかのケースでトリガーされます。

- チェックインスクリプトがパッケージのダウンロードを試行したときにパッケージが見つからない - アップグレードパッケージを解凍できないか、パッケージ内のインストーラを検証できません。
- OS の問題または依存関係によるインストールプロセスの失敗。

Windows

- CA ルート証明書が存在しない：証明書の問題
- エージェントが最初に MSI インストールパッケージを使用して手動でインストールされた場合は、ユーザーガイドの「プラットフォームが現在サポートされているかどうかを確認する」で、Windowsエディションがサポートされているプラットフォームのリストと一致するかどうかを確認します。
- OS が Windows インストーラ操作用に正しく構成されていることを確認します：Windows Installer の問題
- ホストに十分な空きディスク領域があることを確認します。

Linux

- 最後のエージェントのインストール以降にホスト OS がアップグレードされている場合は、ユーザーガイドの「プラットフォームが現在サポートされているかどうかを確認する」で、現在のリリースがサポートされているプラットフォームのリストと一致していることを確認します。
- 前回のインストール以降、必要な依存関係が変更されていないことを確認します。これらの依存関係を再確認するには、-no-install オプションを指定してエージェント インストーラ スクリプトを実行します。
- ホストに十分な空きディスク領域があることを確認します。

AIX

- 前回のインストール以降、必要な依存関係が変更されていないことを確認します。これらの依存関係を再確認するには、-no-install オプションを指定してエージェント インストーラ スクリプトを実行します。
- ホストに十分な空きディスク領域があることを確認します。

コンバート失敗

現在のエージェントタイプが目的のエージェントタイプと一致しないため、コンバートの試行がタイムアウトしました。この問題は、エージェントがパッケージをダウンロードするために check_in を実行したとき、または wss サービスが convert_commnad をエージェントにプッシュできなかったときに、通信の問題が原因で発生する可能性があります。

すべてのプラットフォーム

- ユーザーガイドの「プラットフォームが現在サポートされているかどうかを確認する」で、現在のリリースおよびエージェントタイプが、サポートされているプラットフォームのリストと一致していることを確認します。

機能の変換

エージェントをあるタイプ（優れた可視性など）から別のタイプ（適用など）に変換する能力は、すべてのエージェントが利用できるわけではありません。変換を実行できないエージェントで変換が要求された場合、異常が報告されます。

同期されていないポリシー

エージェントによって最後に報告された現在のポリシー（NPC）バージョンが、クラスターで生成された現在のバージョンと一致しません。これは、エージェントとクラスタ間の通信エラー、エージェントがローカルファイアウォールでポリシーを適用できない、またはエージェントの適用サービスが実行されていないことが原因である可能性があります。

Windows

- 適用モードが WAF の場合、ファイアウォールの有効化、ルールの追加（ルールの保持をオフの状態で）、またはデフォルトアクションの設定を妨げる GPO がホストに存在しないことを確認します。GPO の設定
- ホストとクラスタの間に接続があることを確認します。 SSL のトラブルシューティング
- 生成されたルール数が **2000** 未満であることを確認します。
- WindowsAgentEngine サービスが実行されていることを確認します。*sc query windowsagentengine*
- 利用可能なシステムリソースがあることを確認します。

Linux

- *iptables* および *ipset* コマンドを使用して、iptables および ipset が存在することを確認します。
- ホストとクラスタの間に接続があることを確認します。 SSL のトラブルシューティング
- tet-enforcer プロセスが実行されていることを確認します。*ps -ef | grep tet-enforcer*

AIX

- *ipf -V* コマンドを使用して、ipfilter がインストールされ、実行されていることを確認します。
- ホストとクラスタの間に接続があることを確認します。 SSL のトラブルシューティング
- tet-enforcer プロセスが実行されていることを確認します。*ps -ef | grep tet-enforcer*

フローのエクスポート：Pcap オープン

Secure Workload エージェントが Pcap デバイスをオープンしてフローをキャプチャできない場合、エージェントログにエラーが表示されます。Pcapデバイスが正常に開かれた場合は、次のように報告されます。

Windows ログ：*C:\Program Files\Cisco Tetration\Logs\TetSen.exe.log*

```
I0609 15:25:52.354 24248 Started capture thread for device <device_name>
I0609 15:25:52.354 71912 Opening device {<device_id>}
```

Linux ログ：*/usr/local/tet/logs/tet-sensor.log*

```
I0610 03:24:22.354 16614 Opening device <device_name>
[2020/06/10 03:24:23:3524] NOTICE: lws_client_connect_2: <device_id>: address 172.29.
↪136.139
```

フローエクスポート：HTTPS 接続

エージェントとクラスタ間の接続は外部でブロックされているため、フローやその他のシステム情報が配信されません。これはホスト上のネットワークファイアウォール、SSL復号化サー

ビス、またはサードパーティのセキュリティエージェントに関する1つ以上の設定の問題が原因で発生します。

- エージェントとクラスタの間に既知のファイアウォールまたはSSL復号化セキュリティデバイスがある場合は、すべての Secure Workload コレクタと VIP の IP アドレスへの通信が許可されていることを確認してください。オンプレミスクラスタの場合、コレクタのリストは、Secure Workload Web インターフェイスの左側にあるナビゲーションバーの、**[トラブルシューティング（Troubleshoot）] > [仮想マシン（Virtual Machines）]** の下に表示されます。collectorDatamover-* を参照してください。Secure Workload クラウドの場合、許可する必要があるすべての IP アドレスがポータルにリストされます。
- SSL 復号化があるかどうかを判断するために、openssl s_client を使用して接続を確立し、返された証明書を表示できます。チェーンに証明書が追加されると、エージェントのローカル CA によって拒否されます。 SSL のトラブルシューティング

証明書の問題

Windows

MSI インストーラの証明書の問題

MSI インストーラは、コード署名証明書を使用して署名されています。

MSI インストーラの場合、バージョン 3.6.x 以降および 3.5.1.31 以降

- リーフ証明書：Cisco Systems, Inc
- 中間証明書：DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1
- ルート証明書：DigiCert Trusted Root G4

MSI インストーラの場合、以前のバージョン

- リーフ証明書：Cisco Systems, Inc
- 中間証明書：Symantec Class 3 SHA256 Code Signing CA
- ルート証明書：VeriSign Class 3 Public Primary Certification Authority - G5

タイムスタンプ証明書を使用します。

MSI インストーラの場合、バージョン 3.6.x 以降および 3.5.1.31 以降

- リーフ証明書：Symantec SHA256 TimeStamping Signer - G3
- 中間証明書：Symantec SHA256 TimeStamping CA
- ルート証明書：VeriSign Universal Root Certification Authority

MSI インストーラの場合、以前のバージョン

- リーフ証明書：Symantec SHA256 Timestamping Signer - G2

- 中間証明書：Symantec SHA256 Timestamping CA
- ルート証明書：VeriSign Universal Root Certification Authority

MSIインストーラのデジタル署名が無効な場合、Windows Sensorのインストールまたはアップグレードは失敗します。次の場合、デジタル署名は無効です。

- MSI インストーラの署名ルート証明書または MSI インストーラのタイムスタンプルート証明書が「信頼されたルート証明機関」ストアにない
- MSI インストーラの署名ルート証明書または MSI インストーラのタイムスタンプルート証明書が期限切れまたは失効している

問題 1

エージェントのインストールがTetUpdate.exe.logで「Msi署名が信頼されていません。0x800b0109（Msi signature is not trusted. 0x800b0109）」というエラーにより失敗することがある

エージェントのインストールが check_conf_update.log で「TetrationAgentInstallaer.msi は正しく署名されていません。中止します（TetrationAgentInstallaer.msi is not signed properly, aborting）」というエラーにより失敗することがある

対処法

- コマンドプロンプトからコマンド *certmgr* を実行します。
- MSI インストーラの署名ルート証明書または MSI インストーラのタイムスタンプルート証明書が [信頼されたルート証明機関（Trusted Root Certification Authorities）] ストアにあるかどうかを確認します。
- 証明書を [信頼されたルート証明機関（Trusted Root Certification Authorities）] ストアに移動します。

問題 2

Windows Sensor のアップグレードが TetUpdate.exe.log の「Msi 署名が信頼されていません。0x800B010C （Msi signature is not trusted. 0x800B010C）」というエラーにより失敗する

Windows Sensorのアップグレードが、check_conf_update.log CERT_TRUST_STATUS.dwErrorStatus: 0x04000024 の次のエラーで失敗する

CERT_TRUST_STATUS.dwInfoStatus: 0x04000024 SignTool Error: WinVerifyTrust returned error: 0x800B010C

証明書が発行者によって明示的に取り消されています。

対処法

- コマンドプロンプトからコマンド *certmgr* を実行します。

- MSI インストーラの署名ルート証明書または MSI インストーラのタイムスタンプルート証明書が [信頼されたルート証明機関（Trusted Root Certification Authorities）] ストアにあるかどうかを確認します。
- 証明書を [信頼されたルート証明機関（Trusted Root Certification Authorities）] ストアにコピーします。

問題 3

Windows Sensor のアップグレードが TetUpdate.exe.log の「Msi 署名が信頼されていません。0x80096005 （Msi signature is not trusted. 0x80096005）」で失敗する

Windows Sensor のアップグレードが、check_conf_update.log の次のエラーで失敗する

アップグレードパッケージの検証に失敗しました。” “error code after running check_conf_update = 16” を終了します。

または

signtool verify /pa /v TetrationAgentInstaller.msi によってこのエラーが生成されます。

SignTool Error: WinVerifyTrust returned error: 0x80096005

タイムスタンプの署名または証明書、あるいはその両方を検証できなかったか、形式が正しくありません。

対処法

- コマンドプロンプトからコマンド *certmgr* を実行します。
- MSI インストーラの署名ルート証明書および MSI インストーラのタイムスタンプルート証明書が「信頼されたルート証明機関」ストアにあるかどうかを確認します。

証明書が見つからない場合は、他のマシンからインポートします。

証明書をインストールするには、次の手順を実行します。

まず、稼働しているいずれかのサーバーから、証明書 VeriSign Universal Root Certificate Authority をエクスポートします。以下の手順に従います。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼されたルート証明機関（Trusted Root Certification Authorities）] の下の証明書 [VeriSign Universal Root Certification Authority] を右クリックし、[すべてのタスクのエクスポート（All tasksExport）] に移動します。
- エクスポートされた証明書を非稼働中のサーバーにコピーしてから、証明書をインポートします。

証明書をインストールするには、次の手順を実行します。

まず、稼働しているいずれかのサーバーから、証明書 VeriSign Universal Root Certificate Authority をエクスポートします。以下の手順に従います。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート認証局（Trusted Root Certification Authorities）] の下にある [証明書（certificates）] タブを右クリックし、[すべてのタスクのインポート（All tasksImport）] に移動します。
- コピーしたルート証明書を選択し、ストアに追加します。

NPCAP インストーラの証明書の問題

Windows 2012、Windows 2012 R2、Windows 8、Windows 8.1 が該当します

NPCAP バージョン：1.55

NPCAP 署名の証明書：

- リーフ証明書：Insecure.Com LLC
- 中間証明書：DigiCert EV Code Signing CA (SHA2)
- ルート証明書：DigiCert High Assurance EV Root CA

NPCAP タイムスタンプ証明書：

- リーフ証明書：DigiCert Timestamp 2021
- 中間証明書：DigiCert SHA2 Assured ID Timestamping CA
- ルート証明書：DigiCert Assured ID Root CA

問題 1

Windows エージェントのインストールに失敗し、msi_installer.log に以下のエラーメッセージが表示されることがある

```
CheckServiceStatus : Exception System.InvalidOperationException: Service npcap was not
found on computer
'.'. ---> System.ComponentModel.Win32Exception: The specified service does not exist as
an installed service
```

対処法

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート認証局（Trusted Root Certification Authority）] ストアの [DigiCert High Assurance EV Root CA] を選択します。
- 証明書が見つからない場合は、他のマシンからインポートします。

証明書をインストールするには、次の手順を実行します。

稼働中のいずれかのサーバーから証明書「DigiCert High Assurance EV Root CA」を最初にエクスポートします。以下の手順に従います。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート証明局（Trusted Root Certification Authorities）] の下にある [DigiCert High Assurance EV Root CA] 証明書を右クリックします。
- エクスポートされた証明書を非稼働中のサーバーにコピーしてから、証明書をインポートします。

証明書をインストールするには、次の手順を実行します。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート認証局（Trusted Root Certification Authorities）] の下にある [証明書（certificates）] タブを右クリックし、[すべてのタスクのインポート（All tasksImport）] に移動します。
- コピーしたルート証明書を選択し、ストアに追加します。

Windows 2008 R2 に適用

NPCAP バージョン：0.991

NPCAP 署名の証明書：

- リーフ証明書：Insecure.Com LLC
- 中間証明書：DigiCert EV Code Signing CA
- ルート証明書：DigiCert High Assurance EV Root CA

NPCAP タイムスタンプ証明書：

- リーフ証明書：DigiCert Timestamp Responder
- 中間証明書：DigiCert Assured ID CA-1
- ルート証明書：VeriSign DigiCert Assured ID Root CA

問題 1

Windows エージェントのインストールに失敗し、msi_installer.log に以下のエラーメッセージが表示されることがある

```
CheckServiceStatus : Exception System.InvalidOperationException: Service npcap was not
found on
computer '.'. —> System.ComponentModel.Win32Exception: The specified service does not
exist as an
installed service
```

対処法

- コマンドプロンプトからコマンド *certmgr* を実行します。

- [信頼できるルート認証局（Trusted Root Certification Authority）] ストアの [DigiCert High Assurance EV Root CA] を選択します。
- 証明書が見つからない場合は、他のマシンからインポートします。

証明書をインストールするには、次の手順を実行します。

稼働中のいずれかのサーバーから証明書「DigiCert High Assurance EV Root CA」を最初にエクスポートします。以下の手順に従います。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート証明局（Trusted Root Certification Authorities）] の下にある [DigiCert High Assurance EV Root CA] 証明書を右クリックします。
- エクスポートされた証明書を非稼働中のサーバーにコピーしてから、証明書をインポートします。

証明書をインストールするには、次の手順を実行します。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート認証局（Trusted Root Certification Authorities）] の下にある [証明書（certificates）] タブを右クリックし、[すべてのタスクのインポート（All tasksImport）] に移動します。
- コピーしたルート証明書を選択し、ストアに追加します。

Windows ホストの名前変更

シナリオ 1：Windows ホストの名前を変更した後、IP アドレスと VRF 情報が表示されない問題を修正する手順：

- TaaS UI から（IP アドレスと VRF 情報が欠落している新しいホスト名を持つ）エントリを削除します。
- Windows ホストから「Cisco Secure Workload エージェント」をアンインストールし、「Cisco Tetration」ディレクトリを削除します（通常、該当するパスは「C:Program FilesCisco Tetration」）。
- Windows ホストに「Cisco Secure Workload エージェント」をインストールします。

上記の手順を実行すると、TaaS UI でエージェントが IP アドレスと VRF 情報を使用して正常に登録されます。

シナリオ 2：計画された Windows ホストの（事前）名前変更の手順：

- Windows ホストから「Cisco Secure Workload エージェント」をアンインストールし、「Cisco Tetration」ディレクトリを削除します（通常、該当するパスは「C:Program FilesCisco Tetration」）。
- Windows ホストの名前を変更して再起動します。

- Windows ホストに「Cisco Agent」をインストールします（新しいホスト名を使用）。Secure Workload

計画されたホストの名前変更に関する上記の手順に従って、エージェントを新しいホスト名で TaaS UI に登録する必要があります。

プラットフォームが現在サポートされているかどうかを確認する

Windows

- コマンド winver.exe を実行します。
- 「サポートされているプラットフォームと要件」に記載されている内容とこのリリースを比較します

Linux

- *cat /etc/os-release* を実行します。
- 「サポートされているプラットフォームと要件」に記載されている内容とこのリリースを比較します

AIX

- コマンド *uname -a* を実行します
- 注：メジャーバージョンとマイナーバージョンが逆になっています。

```
p7-ops2> # uname -a
AIX p7-ops2 1 7 00F8AF944C00
```

- この例では、ホスト名の後の最初の数字がマイナーバージョン、2 番目の数字がメジャーバージョンであるため、AIX バージョン 7.1 になります。「サポートされているプラットフォームと要件」に記載されている内容とこのリリースを比較します

Windows Installer の問題

- *C:\Windows\Installer* ディレクトリが存在することを確認してください。これはファイルエクスプローラには表示されません。最も簡単な確認方法は、CMD セッションで次を実行する方法です。*dir C:\Windows\Installer*
- *Windows Installer* サービスが無効になっていないかどうかを確認します。サービスを手動に設定する必要があります。
- Windows Installer によって他のエラーがレポートされていないかどうかを確認します。**[Windowsログ（Windows Logs）]** > **[アプリケーション（Application）]** > **[送信元（Source）** > **[MsiInstaller]** で Windows システムイベントログを確認します。

必要な Windows サービス

以下は、無効になっている場合にエージェントのインストール問題に関係するサービスのリストです。優れた可視性と適用エージェントの初期インストール時およびアップグレード時に、これらのサービスが稼働中であることを推奨します。

表 *15* : 必要な *Windows* サービス

サービス	インストールの目的
デバイス セットアップ マネージャ	Npcap フィルタドライバーのインストール用のデバイスドライバ管理。
デバイスのインストールサービス	Npcap フィルタドライバのインストールにも使用されます。
Windows インストーラ	エージェントの MSI パッケージのインストールに必要です。
Windows 用ファイアウォール	WAF 適用モードに必要です。
Application Experience	システム上の機能の実行可能ファイルを決定するために使用されます。

（注） アプリケーション体験サービスは、Windows Server 2008、2008R2、2012、2012R2、および Windows 7 のみが対象です。無効にすると、Npcap のインストール中にファイルのロックが発生し、インストールに失敗する可能性があります。

Npcap の問題

Npcap は Windows エージェント専用の PCAP ツールです。エージェントサービスが開始してから 10 秒後に、Npcap のインストールまたはサポートされているバージョンへのアップグレードが試行されます。Npcap サービスのインストールまたはアップグレードが失敗した場合、エージェントは 30 分以内にインストールを再試行します。3 回失敗すると、エージェントは Npcap を以前のサポート対象バージョンにロールバックしようとします（使用可能な場合）。その後、エージェントが Npcap のインストールを試みることはありません。*C:\Program Files\Cisco Tetration\Logs\TetUpdate.exe.log* および *C:\Program Files\Cisco Tetration\Logs\npcap_install.log* を確認して、エラーを特定できます。

Npcap がアップグレードされない（手動またはエージェント経由）

- プロセスが現在 Npcap ライブラリを使用している場合、Npcap は正常にアンインストールされないことがあります。実行状況を調べるには、次のコマンドを実行します。

```
PS C:\Program Files\Npcap> .\NPFInstall.exe -check_dll
WindowsSensor.exe, Wireshark.exe, dumpcap.exe
```

プロセスが一覧表示される場合は、Npcap アップグレードを続行する前にそれらのプロセスを停止する必要があります。Npcap を使用しているプロセスがない場合、上記のコマンドにより <*NULL*> のみが表示されます。

Npcap がインストールされない

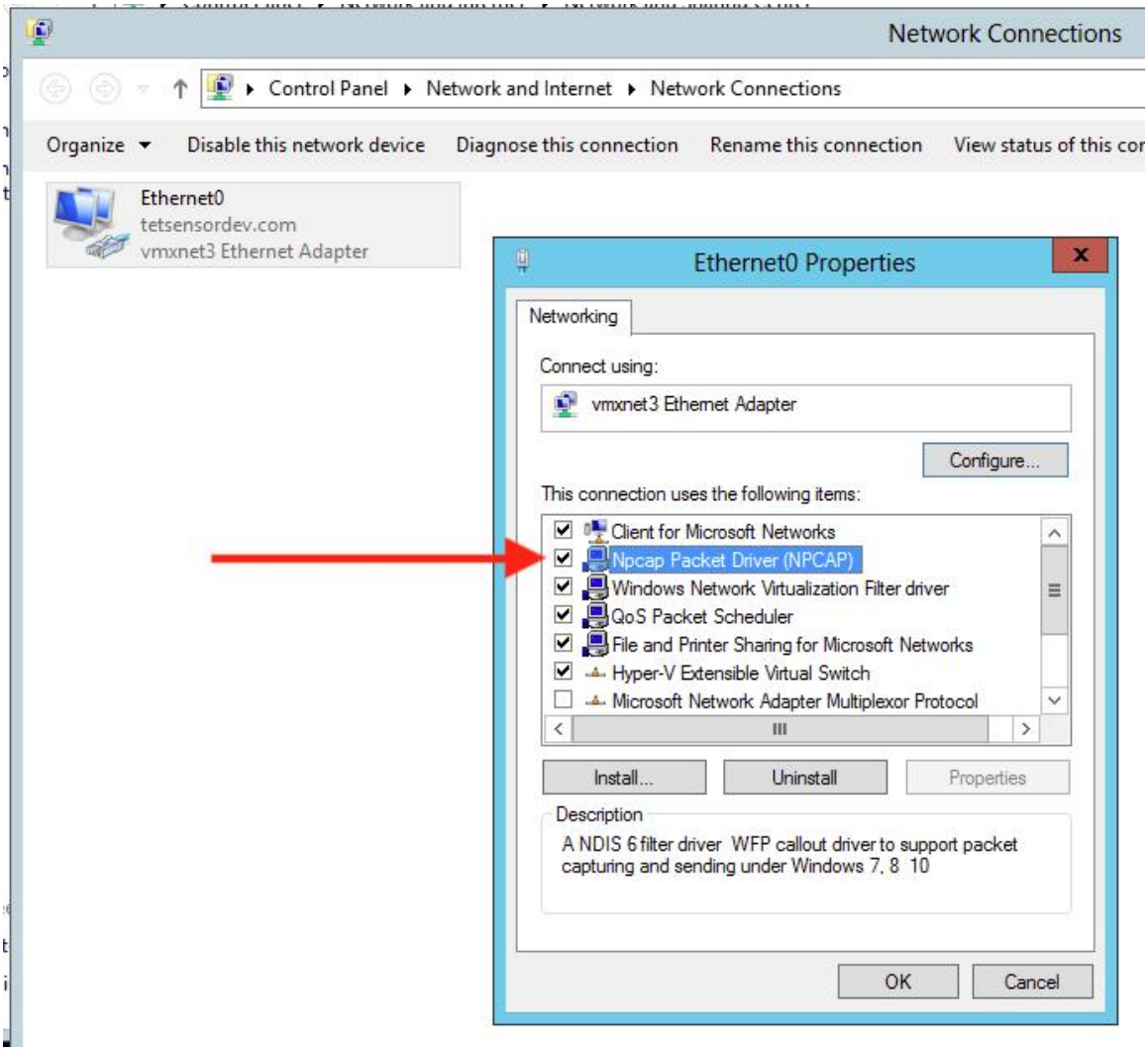
- システムにインストールされている CA 証明書を確認します： NPCAP インストーラの証明書の問題
- Windows インストーラの問題を確認します：Windows Installer の問題
- システム上の他のユーザーがネットワーク インターフェイスに変更を加えていないことを確認します。これが原因で COM ロックが発生し、NDIS ドライバのバインドが妨げられている可能性があります。

Npcap が完全にインストールされているかどうかの確認

手順

ステップ 1 **[コントロールパネル（コントロールパネル）]>[プログラムと機能（Programs and Features）]** をチェックして、Npcap がインストールされているアプリケーションとしてリストされているかどうかを確認します。

ステップ 2 Npcap パケットドライバが問題の NIC にバインドされていることを確認します（チェックマークが表示されます）。



ステップ 3 ドライバが正しくインストールされているかどうか確認します。

```
C:\Windows\system32>pnputil -e | findstr Nmap
Driver package provider : Nmap Project
```

ステップ 4 ドライバサービスがインストールされ、実行中であるかどうかを確認します。

```
C:\Windows\system32>sc query npcap
SERVICE_NAME: npcap
        TYPE : 1 KERNEL_DRIVER
        STATE : 4 RUNNING
```

ステップ 5 レジストリエントリが存在するかどうかを確認します（Npcap がすでに存在することを確認するためにエージェントで使用されます）。

```
C:\Windows\system32>reg query HKLM\software\wow6432node\npcap
HKEY_LOCAL_MACHINE\software\wow6432node\npcap
         AdminOnly REG_DWORD 0x1
         WinPcapCompatible REG_DWORD 0x0
        (Default) REG_SZ C:\Program Files\Npcap
```

ステップ 6 インストールされている Npcap プログラムファイルがすべて存在するかどうかを確認します。

```
C:\Windows\system32>dir "c:\program files\npcap"
 Directory of c:\program files\npcap
04/29/2020 02:42 PM <DIR> .
04/29/2020 02:42 PM <DIR> ..
01/22/2019 08:16 AM 868 CheckStatus.bat
11/29/2016 03:43 PM 1,034 DiagReport.bat
12/04/2018 11:12 PM 8,908 DiagReport.ps1
01/09/2019 09:22 PM 2,959 FixInstall.bat
04/29/2020 02:42 PM 134,240 install.log
01/11/2019 08:52 AM 9,920 LICENSE
03/14/2019 08:59 PM 10,434 npcap.cat
03/14/2019 08:57 PM 8,657 npcap.inf
03/14/2019 09:00 PM 74,040 npcap.sys
03/14/2019 08:57 PM 2,404 npcap_wfp.inf
03/14/2019 09:00 PM 270,648 NPFInstall.exe
04/29/2020 02:42 PM 107,783 NPFInstall.log
03/14/2019 09:01 PM 175,024 Uninstall.exe
         13 File(s) 806,919 bytes
          2 Dir(s) 264,417,628,160 bytes free
```

ステップ 7 .sys ドライバファイルが Windows のドライバフォルダにあるかどうかを確認します。

```
C:\Windows\system32>dir "C:\Windows\System32\Drivers\npcap.sys"
Directory of C:\Windows\System32\Drivers
03/14/2019 09:00 PM 74,040 npcap.sys
                 1 File(s) 74,040 bytes
```

NPCAP のインストールまたはアップグレードの実行中のネットワーク接続の問題

Windows 2016 のみに適用

サードパーティの LWF（ライトウェイトフィルタ）ドライバー（netmon など）がある場合、またはセットアップでチーミングアダプタが構成されており、エージェントの展開中に NPCAP がインストールされている場合、次のような状況が発生する可能性があります。

RDP が再接続される

NetBios サービスが再起動する

同様なネットワーク接続に関する問題

これは、Windows 2016 OS のバグが原因です。

Npcap との NIC チーミングの互換性の問題

チーミング NIC 機能は、物理 NIC（Intel、Broadcom、Realtek、MS 仮想アダプタなど）とチーミングドライバ構成 (スイッチベースでロードバランシングまたはフェールオーバーを備え、複数の NIC にパケットを分散するアルゴリズム) を基盤としています。

一部の NPCAP バージョンでは、特に下位のチーミング NIC へのバインド時に、チーミング NIC との互換性の問題があります。

現在の Secure Workload センサーソフトウェアは、Microsoft がサポートする NIC チーミングを使用してテストされています。

```
NIC type : Intel(R) 82574L Gigabit Network Connection
Teaming Mode : Switch Independent
Load Balancing Mode: Address Hash
OS : Windows 2012 , Windows 2012 R2, Windows 2016, Windows 2019
NPCAP version: 1.55

.
```



（注） Windows 2008R2 は、Microsoft がサポートする NIC チーミングに対応していません。

ネットワークフローを報告しない VDI インスタンス VM

TetSensor サービスは、NPCAP サービスが実行されているときに、複製された VM のネットワークフローをキャプチャしないことがあります。この問題は、MSI インストーラを使用して [nostart] フラグなしでエージェントがインストールされた場合、または VM テンプレートかゴールデンイメージで PowerShell インストーラを使用し、[goldenImage] フラグなしでエージェントがインストールされた場合に発生することがあります。

この場合、Secure Workload エージェントサービスは VM テンプレートで実行を開始します。NPCAP がインストールされ、VM テンプレートのネットワークスタックにバインドされます。新しい VM が VM テンプレートから複製されると、NPCAP は新しく複製された VM のネットワークスタックに正しくバインドされません。その結果、NPCAP はネットワークフローをキャプチャできません。

解像度

手順

ステップ 1 **管理者**権限で **cmd.exe** コマンドを実行します。

ステップ 2 **sc stop tetsensor** コマンドを実行して、TetEnforcer サービスを停止します。

ステップ 3 **sc stop tetenforcer** コマンドを実行して、TetEnforcer サービスを停止します。

ステップ 4 **C:\Program Files\Npcap\NPFInstall.exe -check_dll** を実行して、他のアプリケーションが Npcap を使用していないことを確認します。

ステップ 5 Npcap を使用してプロセスを停止します。

ステップ 6 **C:\Program Files\Npcap\NPFInstall.exe -r** を実行してバインディングを再起動します。

ステップ 7 **sc start tetsensor** コマンドを実行して、TetSensor サービスを開始します。

ステップ 8 **sc start tetenforcer** コマンドを実行して、TetEnforcer サービスを開始します。

Npcap でのネットワークパフォーマンス

Windows TetSensor サービスが実行されていると、ネットワークパフォーマンスが影響を受けることが確認されています。Windows TetSensor サービス（tetsen.exe）は、Npcap を使用して

ネットワークフローをキャプチャします。ネットワークフローをキャプチャするNpcapの実装と、tetsen.exeへのネットワークフローが、ネットワークパフォーマンスに影響を及ぼします。

TetSensor をインストール後にネットワークパフォーマンスを比較：クライアント：Windows 2016

Npcap 1.55

TetSensor 構成：適用モード WFP を使用した会話モード

サーバー：Windows 2016

Npcap 1.55

TetSensor 構成：適用モード WFP を使用した会話モード

cmd の実行：iperf3.exe -c<server_ip> -t 40

表 *16 : 121071*：*Npcap 1.55* でのネットワークパフォーマンス

設定	ネットワーク パフォーマンス
TetSensor 未インストール Npcap なし	[ID] インターバル転送帯域幅 [4] 0.00 ～ 40.00 秒 18.2 ギガバイト 3.90 ギガバイト/秒（送信者） [4] 0.00 ～ 40.00 秒 18.2 ギガバイト 3.90 ギガバイト/秒（受信者）
TetSensor インストール済み Npcap インストール済み	[ID] インターバル転送帯域幅 [4] 0.00 ～ 40.00 秒 17.3 ギガバイト 3.72 ギガバイト/秒（送信者） [4] 0.00 ～ 40.00 秒 17.3 ギガバイト 3.72 ギガバイト/秒（受信者）

Npcap 0.9990 でのネットワークパフォーマンス

TetSensor をインストール後にネットワークパフォーマンスを比較：クライアント：Windows 2016

Npcap 0.9990

TetSensor 構成：適用モード WFP を使用した会話モード

サーバー：Windows 2016

Npcap 0.9990

TetSensor 構成：適用モード WFP を使用した会話モード

cmd の実行：iperf3.exe -c<server_ip> -t 40。表：Npcap 0.9990 のネットワークパフォーマンス

class longtable

設定	ネットワーク パフォーマンス
TetSensor インストール済み Npcap インストール済み	[ID] インターバル転送帯域幅 [4] 0.00 ～ 40.00 秒 16.3 ギガバイト 3.50 ギガバイト/秒（送信者） [4] 0.00 ～ 40.00 秒 16.3 ギガバイト 3.50 ギガバイト/秒（受信者）



（注） パフォーマンスは、インストールされている Windows Npcap バージョン、Windows OS、およびネットワーク構成によって異なる場合があります。

OS のパフォーマンスや安定性の問題

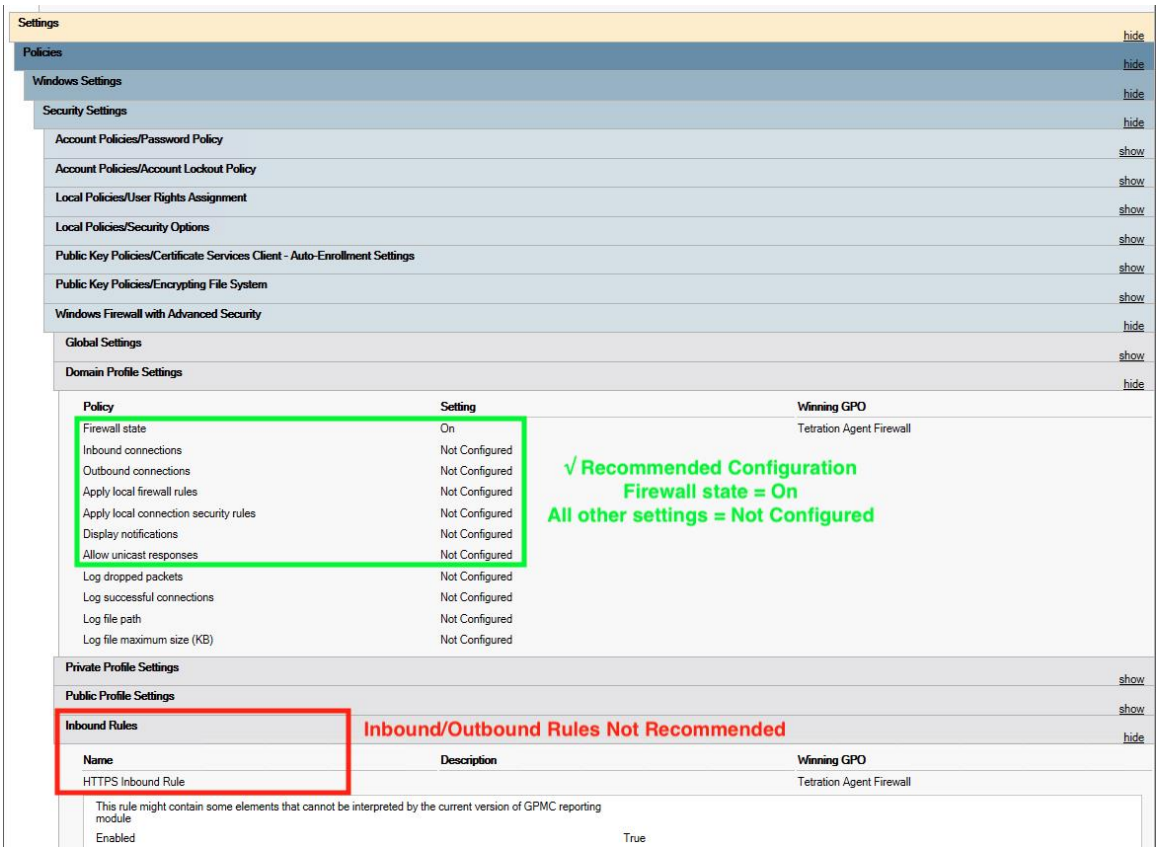
インストールされている NPCAP のバージョンや構成が Secure Workload ソフトウェアでサポートされていない場合、OS で未知のパフォーマンス問題や安定性の問題が発生する可能性があります。

サポートされている NPCAP バージョン：0.991 および 1.55

GPO の設定

ポリシーを適用するエージェントでは、ローカル設定または GPO のいずれかを使用してファイアウォールのみを有効にする必要があります。他のすべての GPO 設定は行わず、「未設定」のままにしておく必要があります。

- GPO 設定が適用をブロックしているかどうかを確認するには、*C:\Program Files\Cisco Tetration\Logs\TetEnf.exe.log* のログを確認し、次のエラー例がないかを探します。
- **「ルールを保持 = いいえ」の設定でルールの競合が発生:**「グループポリシーに設定されているファイアウォールがあります。Secure Workload エージェントには、これを削除する権限がありません。」
- **ファイアウォールがオフに設定されている:**「GPO が DomainProfile のファイアウォールを無効にしています」
- **デフォルトアクションが設定されている:**「グループポリシーは、DomainProfile のデフォルト受信アクションと競合しています」
- ホストに適用されている GPO ポリシーを確認するには、*gpresult.exe /H gpreport.html* を実行し、生成された HTML レポートを開きます。以下の例では、「ルールを保持」が「いいえ」に設定されている場合、*Secure Workload* エージェントのファイアウォールが適用する受信ルールは、適用エージェントと競合します。



クラスタ通信へのエージェント

Secure Workload エージェントは、複数のチャネルを介してクラスタへの接続を維持します。エージェントの種類によって、接続数は異なります。

接続のタイプ

- **WSS**：クラスタへのポート 443 を介した永続的なソケット接続
- **チェックイン**：15 ～ 20 分ごとにクラスタへの HTTPS コールを行い、現在の構成を確認し、更新を確認し、クラスタに対してエージェントのアクティブ状態を更新します。これは、アップグレードの失敗も報告します。
- **フローエクスポート**：フローメタデータをクラスタに送信するための、ポート 443（TaaS）または 5640（オンプレミス）を介した永続的な SSL 接続
- **適用**：ポート 443（Taas）または 5660（オンプレミス）を介した永続的な SSL 接続により、適用ポリシーを取り込み、適用状態を報告します。

接続状態の確認

Teration UI は、非アクティブな（チェックインしなくなった）エージェント、（[統計（Stats）] の [エージェントワークロードプロファイル（Agent Workload Profile ）] ページにある）エクスポートされていないフロー、または失敗した適用のいずれかを報告します。エラーに応じてワークロードのさまざまなログを確認し、問題の原因を特定することができます。

非アクティブなエージェント

Windows ログ：C:\Program Files\Cisco Tetration\Logs\TetUpdate.exe.log

Windows ログ：*C:\Program Files\Cisco Tetration\Logs\check_conf_update.log*

Linux ログ：*/usr/local/tet/logs/check_conf_update.log*

HTTP 応答コード 304 が想定されており、設定の変更がないことを意味します。エラーコード = 2 も同様に想定されます。その他の HTTP 応答コードは、Secure Workload クラスタ上の WSS サービスと通信する際の問題を示します。

```
Tue 06/09/2020 17:25:25.08 check_conf_update: "curl did not return 200 code, it's 304,
‹→ exiting"
Tue 06/09/2020 17:25:25.08 check_conf_update: "error code after running check_conf_
‹→update = 2"
```

- [304] 想定されており、設定の変更はありません。チェックインに成功しました
- [401] 登録に失敗しました。アクティベーションキー（TaaS） がありません
- [403] エージェントはすでに同じ UUID でクラスタに登録されています
- [000] SSLでの接続の問題を示します。curl が WSS サーバーに到達できなかったか、証明書に問題があります。SSLのトラブルシューティングで、SSLのトラブルシューティングを確認してください。

エクスポートされていないフロー

Windows ログ：*C:\Program Files\Cisco Tetration\Logs\TetSen.exe.log*

Linux ログ：*/usr/local/tet/logs/tet-sensor.log*

以下は WSS への接続が成功したことを示します

```
cfgserver.go:261] config server: StateConnected, wss://<config_server_ip>:443/wss/
‹→<sensor_id>/forensic, proxy:
```

以下はコレクタへの接続が成功したことを示します

```
collector.go:258] next collector: StateConnected, ssl://<collector_ip>>:5640
```

WSS またはコレクタへの接続中にエラーが発生した場合は、ファイアウォールの設定を確認するか、エージェントと Secure Workload の間で SSL 復号化が行われていないかどうかを確認してください。 SSL のトラブルシューティング を参照してください。

ポリシー適用の失敗

Windows ログ：*C:\Program Files\Cisco Tetration\Logs\TetEnf.exe.log*

Linux ログ：*/usr/local/tet/logs/tet-enforcer.log*

```
ssl_client.cpp:341] Successfully connected to EFE server
```

EFE サーバーへの接続中にエラーが発生した場合は、ファイアウォールの設定を確認するか、エージェントと Secure Workload の間で SSL 復号化が行われていないかどうかを確認してください。SSL のトラブルシューティング を参照してください。

SSL のトラブルシューティング

エージェント通信の概要

Secure Workload エージェントは、TLS を使用して Secure Workload Cloud SaaS サーバーへの TCP 接続を保護します。これらの接続は、次の 3 つの特徴的なチャネルに分類されます。

- エージェント -> ポート TCP/443（TLS）（sensorVIP）経由の Cisco Secure Workload SaaS 制御チャネル

  これは、エージェントが Secure Workload に登録できるようにする低ボリュームの制御チャネルであり、設定のプッシュとソフトウェアアップグレード通知も処理します。

- エージェント -> TCP/443（TLS）（コレクタ）経由の Cisco Secure Workload SaaS フローデータ

  フローデータは、抽出されたフローメタデータ情報であり、このデータは一度に 16 個の IP アドレスのセット 1 つに送信されます。2 番目の IP アドレスのセットはスタンバイ用です。これは、実際のサーバートラフィックの約 1 ～ 5% を占めます。

- エージェント -> TCP/443（TLS）（efe）を介した Cisco Secure Workload SaaS 適用データ

  適用データチャネルは、ポリシーをセンサーにプッシュし、適用統計を収集するために使用される低ボリュームの制御チャネルです。

センサーは、エージェントとともにインストールされているローカル CA に対して、Secure Workload クラウドの制御、データ、および適用サーバーからの TLS 証明書を検証します。他の CA は使用されないため、エージェントに送信される他の証明書は検証に失敗し、エージェントは接続されません。結果的に、エージェントは登録、チェックイン、フローの送信、または適用ポリシーの受信を行わなくなります。

エージェント通信の IP トラフィックの構成

多くの場合に使用される一般的な構成は、エージェント（ワークフロー）と Secure Workload TaaS の間に境界ファイアウォールと、場合によってはプロキシを配置することです。

（注） Secure Workload は、オンボーディング中にもゲートウェイ/NAT IP 情報を収集し、テナント作成時に情報を自動的に追加します。ポータルで新しい IP アドレスを追加するか IP アドレスを変更する場合、変更には Secure Workload スタッフによる確認と承認が必要です。

TaaS ポータルでゲートウェイ/NAT IP アドレスを追加することに加えて、アウトバウンドトラフィックと変更されていないトラフィックを許可するために、ネットワークにさらに変更が必要になる場合があります。

境界ファイアウォールで TLS/HTTPS を介したアウトバウンドポート 443 を許可する

復号 Web プロキシが使用されている場合は、Web プロキシでプロキシバイパスと SSL/TLS バイパスを構成します。

（注） データセンターで透過的な Web プロキシを使用している場合は、特定の SaaS IP アドレスをルーティングし、バイパスルールを構成する必要があります。センサーは、自動 HTTPS リダイレクトを実行できない接続です。

エージェントの通信先 IP のリストは、TaaS ポータルで入手できます。ファイアウォールのアウトバウンド構成とプロキシバイパスに追加する IP には、collector-n、efe-n（適用が展開されている場合のみ）、および sensorVIP というラベルが付けられます。通常、エージェント通信用に追加する IP は 17 ～ 33 個ありますが、TaaS 構成によってはそれ以上または以下になる可能性があります。

SSL/TLS 接続のトラブルシューティング

前のセクションで説明したとおり、エージェント通信の SSL/TLS 復号化をバイパスするために、明示的または透過的な Web プロキシを設定することが重要です。バイパスが設定されていない場合、これらのプロキシは復号化を試みる可能性があります

SSL/TLS トラフィックは、自身の証明書をエージェントに送信することによるトラフィックです。エージェントはローカル CA のみを使用して証明書を検証するため、これらのプロキシ証明書によって接続エラーが発生します。

症状には、エージェントがクラスタに登録できない、エージェントがチェックインしない、エージェントがフローを送信しない、（適用が有効になっている場合）エージェントが設定の適用を受信しないなどがあります。

（注） 以下のトラブルシューティング手順は、デフォルトのインストールパスが使用されたことを前提としています。Windows: C:Program FilesCisco Tetration Linux: /usr/local/tet. エージェントを別の場所にインストールした場合は、手順をその場所に置き換えてください。

SSL/TLS 接続の問題は、エージェントログで報告されます。ログに SSL エラーがあるかどうかを確認するには、観察された関連する問題に対して次のコマンドを実行します。

登録、チェックイン

Linux

```
grep "NSS error" /usr/local/tet/log/check_conf_update.log
```

Windows（PowerShell）

```
get-content "C:\Program Files\Cisco Tetration\logs\check_conf_update.log" | select-
‹→string -pattern "cSSL Certificate problem"
```

```
get-content "C:\Program Files\Cisco Tetration\logs\TetUpdate.exe.log" | select-
‹→string -pattern "curl failed SSL peer certificate"
```

フロー（Flows）

報告される SSL/TLS 接続問題のほとんどは、エージェントの最初の接続および登録中に発生します。フローを送信するには、接続を試みる前に登録が完了している必要があります。ここで表示される SSL/TLS エラーは、センサーの VIP IP は許可されているが、コレクタの IP が許可されていないことが原因です。

Linux

```
grep "SSL connect error" /usr/local/tet/log/tet-sensor.log
```

Windows（PowerShell）

```
get-content "C:\Program Files\Cisco Tetration\logs\WindowsSensor*.log" | select-
‹→string -pattern "Certificate verification error"
```

施行

Linux

```
grep "Unable to validate the signing cert" /usr/local/tet/log/tet-enforcer.log
```

Windows（PowerShell）

```
get-content "C:\Program Files\Cisco Tetration\logs\WindowsSensor*.log" | select-
‹→string -pattern "Handshake failed"
```

上記のログチェックにSSLエラーが表示された場合は、次のコマンドを使用して、エージェントに送信されている証明書を確認できます。

Explicit Proxy - where a proxy is configured in user.cfg

Linux

```
curl -v -x http://<proxy_address>:<port> https://<sensorVIP>:443
```

Windows（PowerShell）

```
cd "C:\Program Files\Cisco Tetration"
.\curl.exe -kv -x http://<proxy_address>:<port> https://<sensorVIP>:443
```

透過型プロキシ：user.cfg プロキシ設定は不要です。このプロキシは、エージェントからインターネットへのすべての HTTP（S）トラフィック間で設定されたプロキシです。

Linux

```
openssl s_client -connect <sensorVIP from TaaS Portal>:443 -CAfile /usr/local/tet/
‹→cert/ca.cert
```

Windows（PowerShell）

```
cd C:\Program Files\Cisco Tetration
.\openssl.exe s_client -connect <sensorVIP from TaaS Portal>:443 -CAfile cert\ca.cert
```

openssl s_client respose で次のものを探しています

```
Verify return code: 0 (ok)
```

エラーが表示された場合は、証明書を調べてください。証明書（チェーン）の例には、次の証明書のみを含める必要があります（CN IP は一例です）。

証明書チェーン

```
0 s:/C=US/ST=CA/L=San Jose/O=Cisco Systems, Inc./OU=Tetration, Insieme BU/CN=129.146.
‹→155.109
i:/C=US/ST=CA/L=San Jose/O=Cisco Systems, Inc./OU=Tetration Analytics/CN=Customer CA
```

追加の証明書が表示される場合は、エージェントと Cisco Secure Workload の間に Web 復号化プロキシがある可能性があります。セキュリティグループまたはネットワークグループに連絡し、上記の「エージェント通信の IP トラフィックの設定」セクションでリストされている IP を使用して、プロキシバイパスが設定されていることを確認してください。

Windows 2016 サーバーで Windows Sensor のインストールスクリプトが失敗する：「基になる接続が切断されました。受信時に予期しないエラーが発生しました」というエラーメッセージが表示される場合があります。考えられる理由は、PowerShell で設定されている SSL/TLS バージョンである可能性があります。

実行されている SSL/TLS バージョンを調べるには、次のコマンドを実行します。

```
[Net.ServicePointManager]::SecurityProtocol
```

上記のコマンドの出力が次である場合：

```
Ssl3, Tls
```

次に、以下のコマンドを使用して許可されたプロトコルを変更し、インストールを再試行してください。

```
[Net.ServicePointManager]::SecurityProtocol = [System.Net.SecurityProtocolType]'Ssl3,
‹→Tls,Tls11,Tls12'
```

エージェントの操作

Q：エージェントを正常にインストールしましたが、[UIセンサーモニタリング（UI Sensor Monitoring）] ページに表示されません。

A：エージェントが動作を開始する前に、クラスタ内で実行されているバックエンドサーバーにエージェントを登録する必要があります。エージェントがUIページに表示されない場合は、登録に失敗したことが原因であると考えられます。いくつかのポイントをチェックすることで、登録に失敗した理由を確認できます。

- エージェントとバックエンドサーバー間の接続が正しく機能しているかどうかを確認します。
- curl リクエストをバックエンドサーバーに正しく送信できるかどうかを確認します。

- HAProxyアクセスとバックエンドサーバーのログをチェックして、登録リクエストがサーバーに届いたかどうかを確認します。
- ログファイルで curl リクエストから返されたエラーを確認します。

Q：エージェントがインストールされ、UI ページでエージェントを確認できましたが、[ソフトウェアバージョン（SW Ver）] 列に、バージョンを示す文字列ではなく [初期化中（initializing）] と表示されます。

A：エージェントを最初にインストールしてバックエンドサーバーに登録した後、エージェントがバージョンを報告するようになるまでさらに 30 分かかります。

Q：エージェントは適切にアップグレードされていますが、[ソフトウェアバージョン（SW Ver）] フィールドには長時間にわたり（数時間など）古いバージョンが表示されたままになっています。

A：エージェントが正常にアップグレードされると、エージェントは curl リクエストを送信して現在実行中のバージョンを報告し、同じリクエストで新しいバージョンがあるか確認しようとします。次のようないくつかの理由により、リクエストがバックエンドに到達できなかった可能性があります。

- リクエストがタイムアウトし、時間内に応答を取得できませんでした。
- ネットワークに問題が発生し、エージェントがバックエンドサーバーに接続できませんでした。

Q：RHEL/CentOS-6.x でエージェントを実行し、正常に動作しています。OS を RHEL/CentOS-7.x にアップグレードする予定です。アップグレード後もエージェントは動作しますか？

A：現在、OS をアップグレードするシナリオ（特にメジャーリリースのアップグレード）はサポートされていません。OS のアップグレード後にエージェントを動作させるには、次の手順を実行します。

- 既存のエージェントソフトウェアをアンインストールします。
- 証明書を含むすべてのファイルをクリーンアップします。
- UI に移動し、エージェントエントリを削除します。
- OS を目的のバージョンにアップグレードします。
- 新しい OS にエージェントソフトウェアをインストールします。

Q：RHEL/CentOS-6.x でエージェントを実行し、正常に動作しています。ホストの名前を変更する予定です。名前の変更/再起動後もエージェントは動作しますか？

A：エージェント ID は、ホスト名と bios-uuid を含むホストの一意性に基づいて計算されます。ホスト名を変更すると、ホストの ID が変更されます。次の操作を実行することをお勧めします。

- 既存のエージェントソフトウェアをアンインストールします。
- 証明書を含むすべてのファイルをクリーンアップします。

- UI に移動し、古いエージェントエントリを削除します。
- Windows ホストの名前を変更して再起動します。
- エージェントソフトウェアを再インストールします。

Q：Windows ホストで、ルールの追加/削除/変更によってファイアウォールの逸脱が発生しました。ルールを探すにはどうすればよいですか？

A：逸脱が検出されると、エージェントはファイアウォールイベントの最後の 15 秒間を「C:\Windows\System32\config\systemprofile\AppData\Roaming\tet\firewall_events」に記録します。逸脱の原因となったルールは、policy_dev_<policy id>_<timestamp>.txt として作成された最後のファイルで見つかります。

Q：Windows ホストにエージェントを正常にインストールしましたが、センサーからのフローの報告が表示されません。なぜですか？

A：Windows ホストでフローを収集するには、Npcap が必要です。Npcap は、エージェントが正常にインストールされてから 10 秒後にインストールされます。数分経ってもセンサーがフローを報告しない場合は、エージェントとバックエンドサーバー間の接続が正しく機能しているかどうか、および Npcap が正しくインストールされているかどうかを確認してください。Npcap の問題

Q：Windows ホスト（2008 R2）にエージェントを正常にインストールしましたが、tetsensor サービスの実行中にシステムクロックがドリフトします。なぜですか？

A：これは、Go および Windows 2008 R2 の既知の問題です。詳細については、「Golang および Win2008 R2」を参照してください。

tetsensor サービスの一部として実行されるプロセス tet-main.exe は、Go バージョン 1.15 を使用して構築されています。そのため、tetsensor サービスの実行中にシステムクロックがドリフトします。

この問題は、Windows 2008 R2 ワークロードが外部 NTP サーバーまたはドメインコントローラを NTP サーバーとして使用するように設定されている場合に発生します。

考えられる回避策：

1. NTP に定期的にクロックを同期させます：w32tm /resync /force
2. tet-main.exe を手動で無効にします。
   - 「管理者」権限で cmd.exe を実行します。
   - regedit.exe を実行します。
   - 「HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\TetSensor」に移動します。
   - 「ImagePath」をダブルクリックします。
   - 値を編集し、tet-main.exe を削除します。

     ビフォー：“C:\Program Files\Cisco Tetration\TetSenEngine.exe” TetSensor TetSen.exe “-f sensor_config” tet-main.exe ” ” TetUpdate.exe

アフター：“C:\Program Files\Cisco Tetration\TetSenEngine.exe” TetSensor TetSen.exe “-f sensor_config” TetUpdate.exe

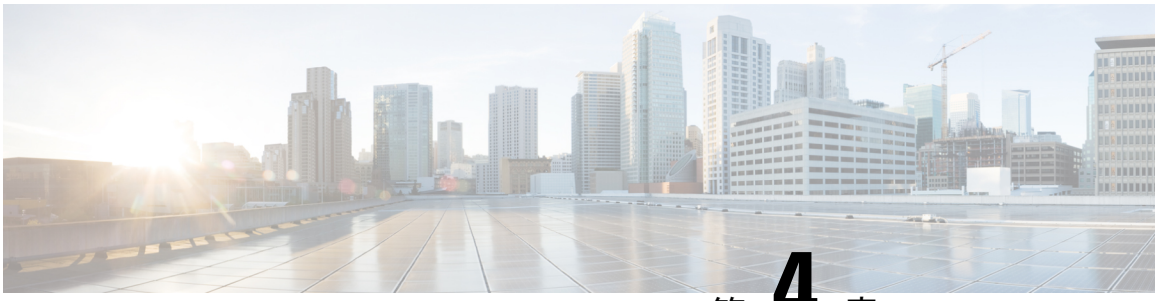
- tetsensor サービスを再起動します。



（注） エージェントがアップグレードされるたびに、tet-main.exe を無効にしてください。

3. 外部 NTP サーバー設定を削除します。
   - コマンドを実行します：w32tm /config /update /manualpeerlist: /syncfromflags:manual /reliable:yes
   - Windows タイムサービス、W32Time を再起動します。

この問題の詳細については、「CSCwb8009」を参照してください。



第 4 章

外部オーケストレータ

外部オーケストレータを使用して、ネットワーク上のシステムからワークロードに関する既存のメタデータを収集できます。一部の外部オーケストレータは、セグメンテーションポリシーを適用することもできます。

ワークロードラベルによるレコード承認システムが存在する環境の場合は、外部オーケストレータと連携してラベルを自動的にインポートする方法を用意しています。レコードシステムの変更は、Secure Workload によって自動的に学習され、インベントリのラベルを更新するために使用されます。ラベルの機能と用途の詳細については、「ワークロードラベル」を参照してください。

現在サポートされている外部オーケストレータ：

表 *17*：現在サポートされている外部オーケストレータ

タイプ	説明/用途
VMware vCenter	ホスト名、IP アドレス、ラベルなどの仮想マシンデータを vCenter サーバーから Secure Workload インポートする際に使用します。生成されたラベルを使用して、Secure Workload の範囲と適用ポリシーを作成できます。
Amazon Web Services	（新しい AWS オーケストレータを作成することはできません。代わりに、AWS コネクタを作成します。「AWS コネクタ」を参照してください。既存の AWS オーケストレータは読み取り専用です)。ホスト名、IP アドレス、ラベルなどの EC2 サーバーインスタンスのデータを特定の AWS アカウントから Secure Workload にインポートする際に使用します。生成されたラベルを使用して、Secure Workload の範囲とポリシーを作成できます。

タイプ	説明/用途
Kubernetes/OpenShift	ノード、ポッド、サービス、ラベルなどの Kubernetes のエンティティをインポートする際に使用します。これらのラベルを Secure Workload で使用して、範囲とポリシーを定義できます。
DNS	ゾーン転送により DNS サーバーから A/AAAA や CNAME レコードをインポートする際に使用します。これにより、Secure Workload の範囲とポリシーを定義する際に便利な DNS 名がラベルとして生成されます。
Infoblox	IPAM または DNS が有効になっている Infoblox アプライアンスから、拡張可能な属性を持つネットワーク、ホスト、および A/AAAA レコードをインポートする際に使用します。インポートされた拡張可能な属性は、Secure Workload の範囲およびポリシーのラベルとして使用できます。
F5 BIG-IP	特定の F5 ロードバランサから仮想サーバー構成を読み取り、提供サービスのラベルを生成します。このラベルは、Secure Workload で適用ポリシーを定義するために使用できます。ポリシー適用機能は、F5 REST API を介してそれらを F5 ポリシー規則に変換します。
Citrix Netscaler	特定の Netscaler ロードバランサから仮想サーバー構成を読み取り、提供サービスのラベルを生成します。このラベルは、Secure Workload で適用ポリシーを定義するために使用できます。ポリシー適用機能は、Netscaler の REST API を介してポリシーを Netscaler ACLに変換します。
Cisco Secure Firewall Management Center *(BETA)*	REST API を使用して、特定の Cisco Secure Firewall Management Center に登録されているすべての Cisco Secure Firewall Threat Defense（旧称 Firepower Threat Defense または FTD）デバイスにポリシーを展開する際に使用します。

- [外部オーケストレータ（External Orchestrators）] ページへの移動 （165 ページ）
- 外部オーケストレータの一覧表示 （165 ページ）
- 外部オーケストレータの作成 （166 ページ）

• 外部オーケストレータの編集 （171 ページ）
• 外部オーケストレータの削除 （171 ページ）
• オーケストレータにより生成されるラベル （171 ページ）
• Secure Connector （172 ページ）
• Amazon Web Services （181 ページ）
• Kubernetes/OpenShift （184 ページ）
• VMware vCenter （192 ページ）
• DNS （195 ページ）
• Infoblox （197 ページ）
• F5 BIG-IP （200 ページ）
• Citrix Netscaler （208 ページ）
• TAXII （212 ページ）
• Cisco Secure Firewall Management Center （215 ページ）

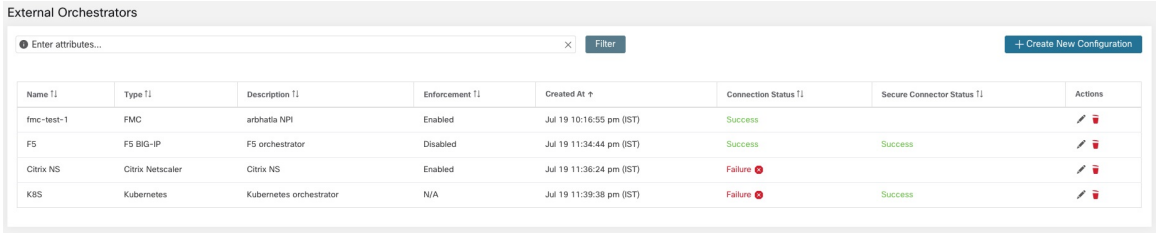
[外部オーケストレータ（External Orchestrators）] ページへの移動

外部オーケストレータのメインページには、左側のメニューバーから **[管理（Manage）]** > **[外部オーケストレータ（External Orchestrators）]** を選択してアクセスできます。

外部オーケストレータの一覧表示

外部オーケストレータのメインページには、既存の外部オーケストレータが表示されます。また、外部オーケストレータを変更および削除する機能と、新しい外部オーケストレータを作成する機能があります。

図 *34* : 外部オーケストレータのメインページ

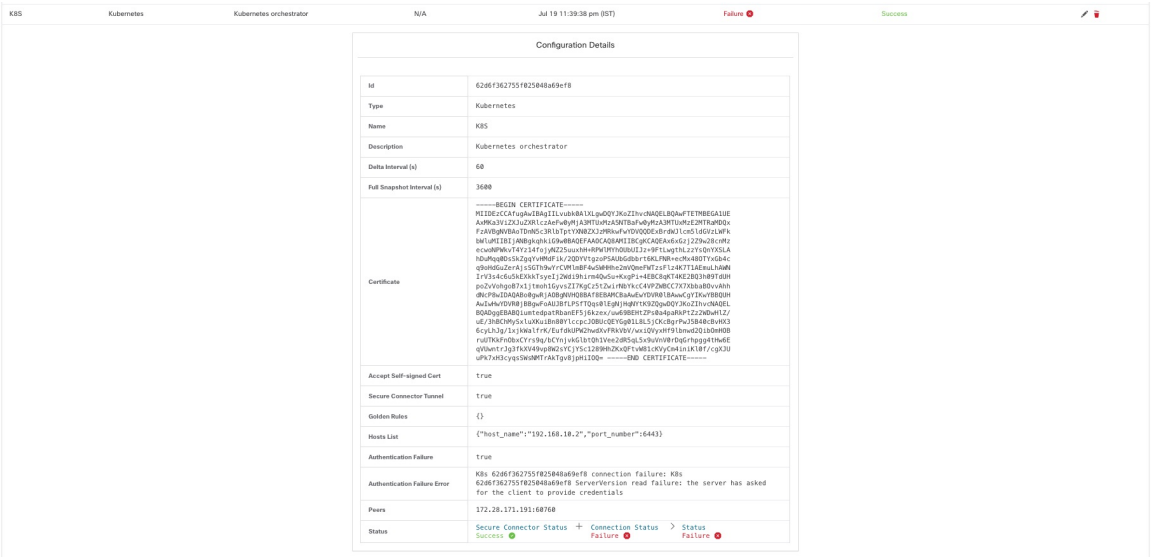


各行には、外部オーケストレータの短いバージョンが、[名前（Name）]、[タイプ（Type）]、[説明（Description）]、[適用（Enforcement）]、[作成日時（Created at）]、[接続ステータス（Connection Status）]、および [Secure Connectorステータス（Secure Connector Status）] とともに表示されます。[接続ステータス（Connection Status）] には、指定された外部データソースへの接続が成功したか失敗したかが示されます。[Secure Connectorステータス（Secure Connector

Status）] には、Secure Connector トンネルのステータス（[成功（Success）] または [失敗（Failure）]）が表示されます。トンネルが有効になっていない場合は、N/A が表示されます。

Secure Connector トンネルは、外部オーケストレータ構成の作成中に有効化できます。Secure Connector トンネルが有効になっている場合、外部オーケストレータの [接続ステータス（Connection Status）] は、認証ステータスと Secure Connector ステータスの両方に依存します。Secure Connector トンネルが有効になっていない場合、外部オーケストレータの [接続ステータス（Connection Status）] は認証ステータスのみに依存します。[成功（Success）] または [失敗（Failure）] のステータスに関係なく、それぞれの行をクリックして詳細を表示できます。Secure Connector クライアントメトリックの詳細を確認するには、[ステータス（Status）] 行をクリックするか、左ペインで**[管理（Manage）] > [Secure Connector]** に移動します。

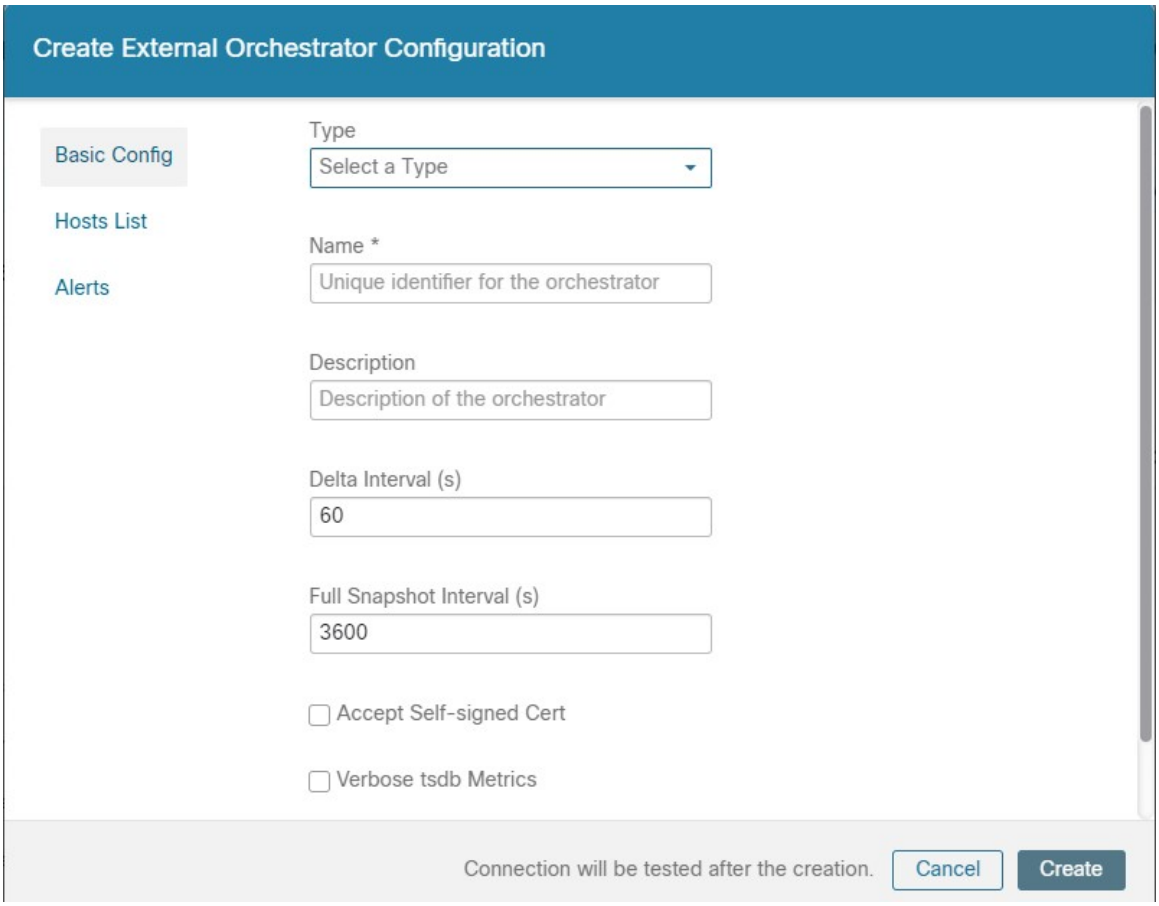
図 *35 :* 外部オーケストレータ認証の失敗例



外部オーケストレータの作成

新しい外部オーケストレータを作成するには、外部オーケストレータのメインページで [新しい設定の作成（Create New Configuration）] ボタンをクリックします。これによりモーダルダイアログが表示され、名前を入力して外部オーケストレータタイプを選択できます。下の図は、基本設定ページを示しています。

図 *36* : 外部オーケストレータ設定の作成



次の表では、外部オーケストレータの共通フィールドについて説明します。選択したタイプに応じて、[基本設定（Basic Config）]ページで追加のパラメータを指定する必要があります。パラメータについては、以下に示す個々の外部オーケストレータのそれぞれのセクションで取り上げます。

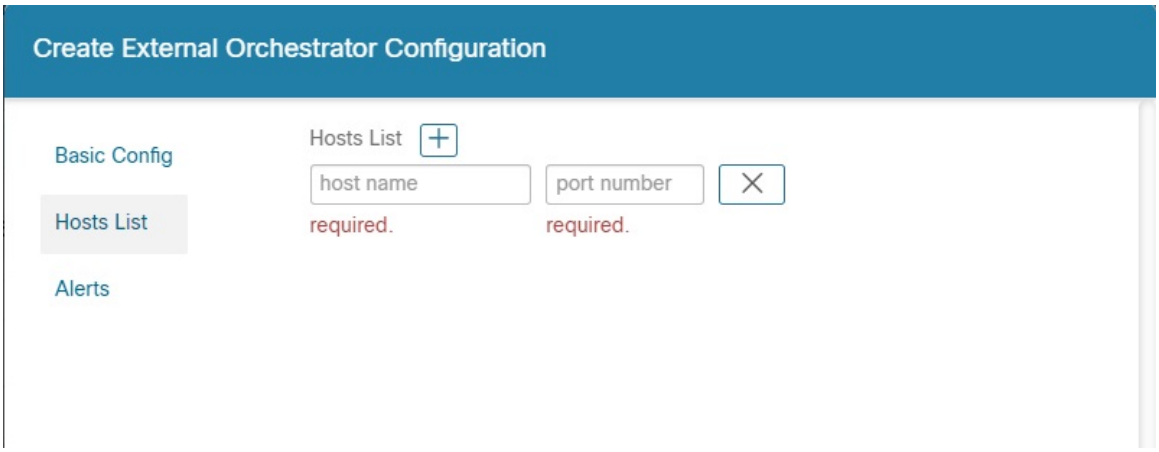
共通フィールド	必須	説明
タイプ	○	リストから外部オーケストレータを選択します。
[名前（Name）]	○	アクティブなテナントに対して一意である必要がある外部オーケストレータの名前。
説明（Description）	×	外部オーケストレータの説明。

共通フィールド	必須	説明
フルスナップショット間隔（Full Snapshot Interval(s)）	○	外部オーケストレータが、選択した [タイプ（Type）] から設定の完全なスナップショットをインポートしようとする間隔（秒）。
自己署名証明書の受け入れ（Accept Self-signed Cert）	×	選択した [タイプ（Type）] から設定データを取得するために、Secure Workload で使用される HTTPS 接続の自己署名サーバー証明書を受け入れるには、このオプションをオンにします。デフォルトでは、自己署名サーバー証明書は許可されません。
セキュアコネクタトンネル（Secure Connector Tunnel）	×	Secure Workload クラスタへの接続が Cisco Secure Connector トンネルを介してトンネリングされるように設定するには、このオプションをオンにします。

（注） 上の図に示されている [デルタ間隔（Delta interval）] および [詳細なTSDBメトリック（Verbose TSDB Metrics）] フィールドはオプションであり、以下のそれぞれの説明で記載されている特定の外部オーケストレータのみが対象になります。

外部オーケストレータのタイプが [AWS] である場合を除き、[ホストリスト（Hosts List）] を指定する必要があります。外部オーケストレータがデータを取得してラベルを生成する外部データソースのネットワークアドレスを指定します。これを行うには、次の図に示すように、左側の [ホストリスト（Hosts List）] タブをクリックします。

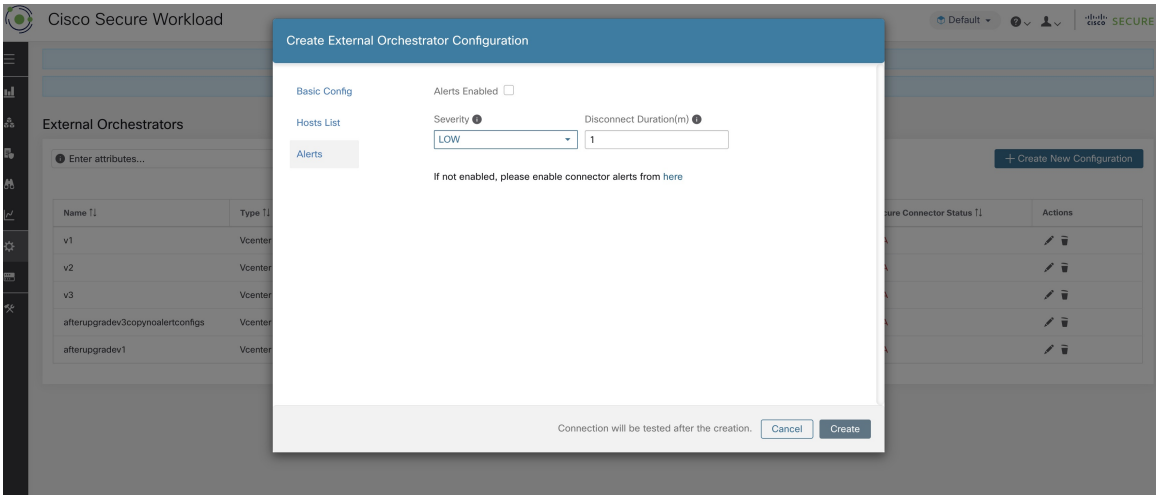
図 *37* : 外部オーケストレータのホストリスト



新しいホストリストエントリを追加するには、プラス記号をクリックします。各行には、有効な DNS ホスト名、IPv4 または IPv6 アドレス、およびポート番号が含まれている必要があります。選択した外部オーケストレータタイプに応じて、高可用性または冗長性の目的で複数のホストを入力できます。詳細については、以下の選択した外部オーケストレータのそれぞれの説明を参照してください。

外部オーケストレータのアラートを設定するには、次の図に示すように、左側の [アラート（Alert）] タブをクリックします。

図 *38* : 外部オーケストレータのアラート



外部オーケストレータごとに、アラートを設定するには、追加のパラメータを指定する必要があります。パラメータについては、以下に示す個々の外部オーケストレータのそれぞれのセクションで取り上げます。

この外部オーケストレータのアラートを有効にするには、[アラート有効化（Alert enabled）] チェックボックスをオンにします。

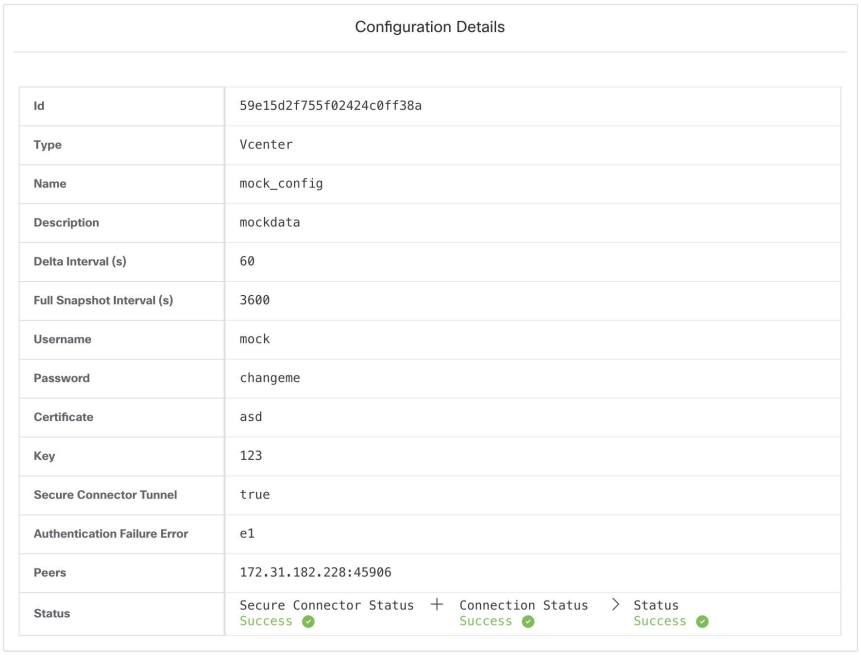
（注） **[管理（Manage）] > [ワークロード（Workloads）] > [アラート設定（Alert Configs）]** ページから、コネクタアラートも有効になっていることを確認します。

外部オーケストレータのアラートを設定するには、[アラートの重大度（Alert Severity）] レベルと分単位の [切断時間（Disconnect Duration）] を選択します。

フィールド	説明
重大度	このルールの重大度レベルを選択します（[低（LOW）]、[中（MEDIUM）]、[高（HIGH）]、[重大（CRITICAL）]、または [即時対応（IMMEDIATE ACTION）]）。
[切断時間（Disconnect Duration）]（分）	接続が切断される時間の長さ。

[作成（Create）] ボタンをクリックして新しい外部オーケストレータを作成します。リストビューの該当する行をクリックすると、設定の詳細を表示できます。

図 *39* : 外部オーケストレータ設定の詳細

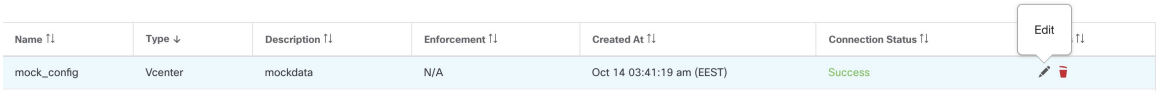


（注） 外部オーケストレータからの最初の完全スナップショットプルは非同期操作であるため、接続ステータスフィールドが更新されるまでに約 1 分かかります。

外部オーケストレータの編集

以下に示すように、外部オーケストレータの行の右側にある鉛筆ボタンをクリックして、構成を変更できる外部オーケストレータを作成する場合と同様のモーダルダイアログを開きます。

図 *40 :* 外部オーケストレータの編集



（注）

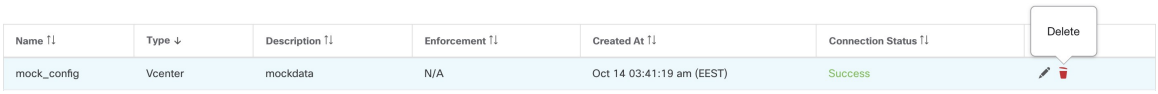
- [タイプ（Type）] フィールドは編集できません。
- 認証にキーと証明書を使用する構成の場合、構成を更新するたびにキーと証明書を提供する必要があります。
- 外部オーケストレータの構成変更は非同期操作であるため、[接続ステータス（Connection Status）] フィールドが更新され、入力された変更が正しいことが確認されるまでに約 1 分かかります。

[更新（Update）] ボタンをクリックして、構成に加えた変更を保存します。

外部オーケストレータの削除

注意 外部オーケストレータを削除すると、そのオーケストレータによって提供されたラベルも削除され、ポリシーに影響します。外部オーケストレータを削除するには、以下に示すようにごみ箱ボタンをクリックします。

図 *41 :* 外部オーケストレータの削除



オーケストレータにより生成されるラベル

Cisco Secure Workload は、次のラベルをすべての AWS インスタンスに追加します。

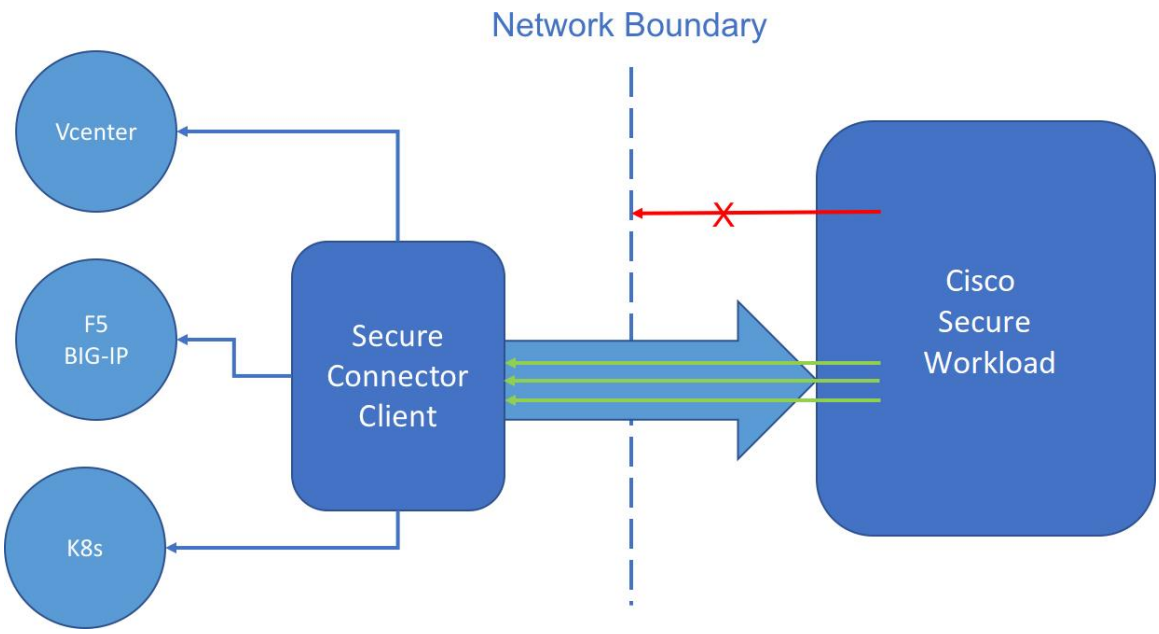
キー	値
orchestrator_system/orch_type	aws

キー	値
orchestrator_system/cluster_name	<*Cluster_name* はユーザーがこのオーケストレータの設定に付けた名前>
orchestrator_system/cluster_id	<|*product*| でのオーケストレータの設定の *UUID*>

Secure Connector

Secure Workload を使用して、ユーザータグをインポートしたり、外部オーケストレータ（「外部オーケストレータ」を参照）にポリシーを適用したりするには、Secure Workload がオーケストレータ API サーバー（vCenter、Kubernetes、F5 BIG-IP など）への発信接続を確立する必要があります。Secure Workload クラスタからオーケストレータへの直接着信接続を許可できない場合があります。Secure Connector は、オーケストレータと同じネットワークから Secure Workload クラスタへの発信接続を確立することにより、この問題を解決します。この接続は、クラスタからの要求をオーケストレータ API サーバーに渡すためのリバーストンネルとして使用されます。

図 *42 : Secure Connector*



ルート範囲ごとに、一度にアクティブにできるトンネルは 1 つだけです。追加のトンネルを開始しようとすると、1 つのトンネルがすでにアクティブであることを示すエラーメッセージが表示されて拒否されます。アクティブトンネルを使用して、クライアントが実行されているネットワークから到達可能な複数のオーケストレータに接続できます。オーケストレータごとの構成は、そのオーケストレータへの接続が Secure Connector トンネルを経由する必要があるかどうかを示すために使用されます。

Secure Connector クライアントと Secure Workload クラスタ間の通信はすべて相互に認証され、TLS を使用して暗号化されます。

セキュリティを向上させるために、適切に保護されている隔離されたマシンに Secure Connector クライアントをインストールすることを推奨します。そのようなマシンには、Secure Workload クラスタへの発信接続のみを許可するファイアウォールルールが必要であり、外部のオーケストレータ API サーバー Secure Workload にはアクセスを許可する必要があります。

Secure Connector トンネルを使用するようにオーケストレータを構成するには、製品の外部オーケストレータを構成する手順を参照してください。

Secure Connector の OpenAPI エンドポイントの詳細については、「Secure Connector API エンドポイント」を参照してください。

技術的な詳細情報

トンネルをブートストラップするために、Secure Connector クライアントが公開キーと秘密キーのペアを作成し、サーバーがリモートで公開キーの証明書に署名します。このリモート署名プロセスを保護し、クライアントが属するルート範囲を識別するために、暗号化された 1 回限りの期間限定トークンが使用されます。サーバー側ではルート範囲ごとに一意の証明書があります。クライアントはこの証明書をサーバーの認証に使用します。証明書は定期的にローテーションされ、通信の機密性が維持されます。

Secure Connector クライアントは、トンネルクライアントと SOCKS5 サーバー内に構築されます。トンネルが開始されると、クライアントは Secure Workload クラスタからのトンネル着信接続を待ちます。着信接続は SOCKS5 サーバーによって処理され、宛先ホストに転送されます。

Secure Connector の要件

Secure Connector クライアントの要件：

- RHEL/CentOS 7（x86_64）
- 2 CPU コアと 4 GB RAM
- Secure Connector を使用するオンプレミス オーケストレータからのデータを処理するのに十分なネットワーク帯域幅
- ポート 443 での Secure Workload クラスタへの発信接続（直接または HTTP(S) プロキシ経由）
- 内部オーケストレータ API サーバーへの発信接続（直接）

Secure Connector クライアントの導入

プロキシ サポート

Secure Connector クライアントは、HTTP(S) プロキシを介した Secure Workload クラスタへの接続をサポートしています。必要に応じて、クライアントの HTTPS_PROXY 環境変数を設定して、プロキシサーバーを設定する必要があります。変数を設定するには、/etc/systemd/system/tetration-secure-connector.service にある systemd サービスファイルの [Service] セクションに次の行を追加します。この設定は、再インストール後は維持されません。スティッキ設定の場合、この行は /etc/systemd/system/tetration-secure-connector.service.d/10-https-proxy.conf の新しいファイルに追加できます。いずれかの設定を有効にするには、*systemctl daemon-reload* を実行して systemd 設定をリロードします。

```
[Service]
Environment="HTTPS_PROXY=<Proxy Server Address>"
```

展開の概要

Secure Connector は、オーケストレータ API サーバーに到達するために、Secure Workload クラスタから内部ネットワークへのリバーストンネルを作成します。

Secure Connector クライアントを起動するには、Secure Connector RPM をダウンロードして、1 回限りの登録トークンを生成する必要があります。

1. サポートされているプラットフォームに 最新の Secure Connector クライアント RPM のダウンロード。
2. 登録トークンの生成。
3. トークンをコピーしてクライアントを開始する、クライアントを起動します。

Secure Connector クライアントの導入

最新の Secure Connector クライアント RPM のダウンロード

手順

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）] > [ワークロード（Workloads）] > [セキュアコネクタ（Secure Connector）]** をクリックします。

ステップ 2 [最新のRPMをダウンロード（Download Latest RPM）] をクリックします。

図 *43* : *[*セキュアコネクタ（***Secure Connector***）*]* ページ

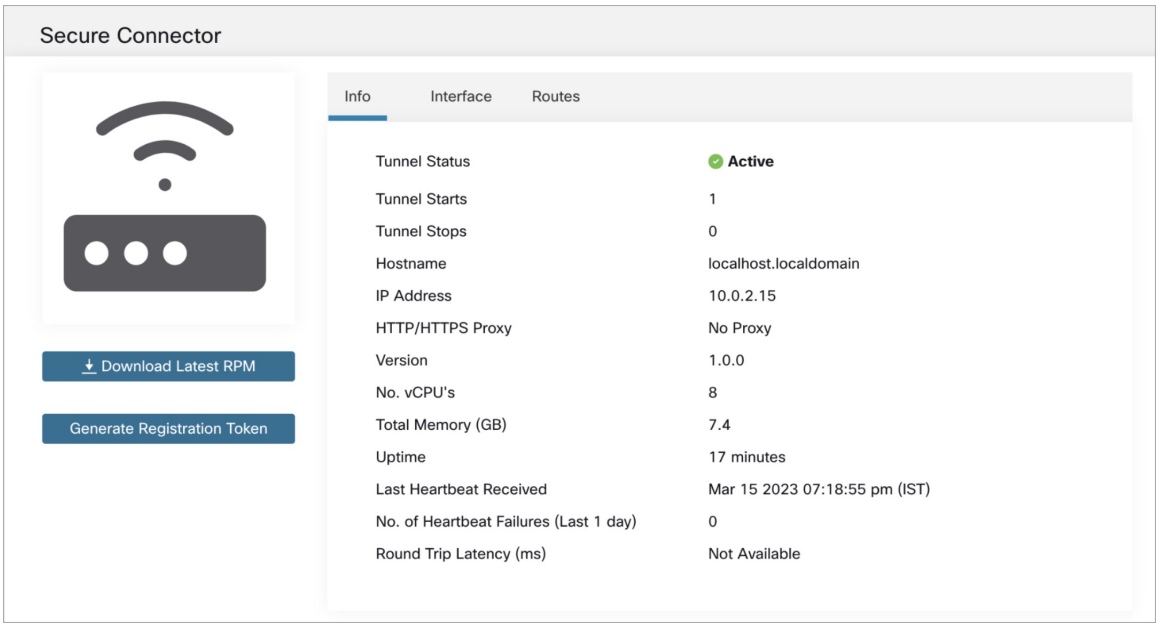
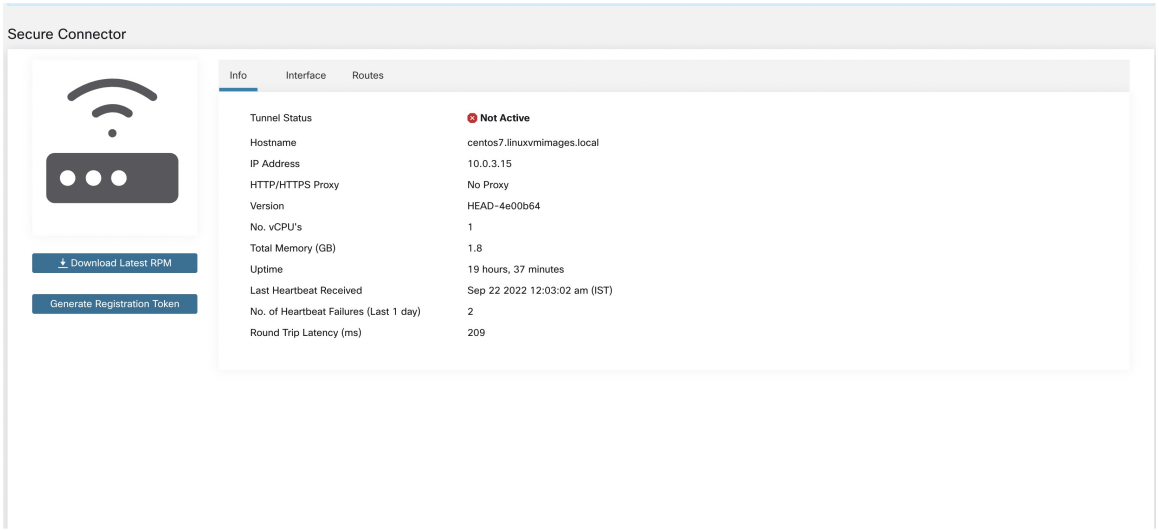


図 *44* : *[*セキュアコネクタ（***Secure Connector***）*]* ページ



ステップ 3 RPM パッケージを展開用の Linux ホストにコピーし、ルート権限で次のコマンドを実行します。

```
rpm -ivh <rpm_filename>
```

登録トークンの生成

手順

ステップ 1 **[管理（Manage）] > [ワークロード（Workloads）] > [セキュアコネクタ（Secure Connector）]** の順にクリックします。

ステップ 2 [登録トークンの生成（Generate Registration Token）] をクリックします。

トークンをコピーしてクライアントを開始する

Secure Connector ページで登録トークンを生成すると、クライアントをブートストラップするための 1 回限りの期間限定トークンを含む *registration.token* ファイルが作成されます。このトークンファイルは、Secure Connector クライアントパッケージをインストールしたホストにコピーする必要があります。トークンファイルをコピーする前に、ホスト上で Secure Connector クライアントが停止していることを確認してください。次のコマンドを使用して、クライアントを停止できます。

```
systemctl stop tetration-secure-connectorsystemctl stop tetration-secure-connector
```

トークンファイルを *cert* フォルダにコピーします。

```
/etc/tetration/cert/registration.token
```

Secure Connector クライアントを再起動します。

```
systemctl start tetration-secure-connector
```

（オプション）特定のバージョンの Secure Connector クライアントを展開する

手順

ステップ 1 **特定のバージョンの Secure Connector クライアント RPM のダウンロード**

a) 左側のナビゲーションバーで、**[管理（Manage）] > [ワークロード（Workloads）] > [エージェント（Agents）]** をクリックします。

b) [インストーラ（Installer）] タブをクリックします。

c) [クラシックパッケージインストーラを使用した手動インストール（Manual Install using classic packaged installers）] を選択し、[次へ（Next）] をクリックします。

Secure Connector クライアントパッケージのエージェントタイプは「Secure Connector」です。

d) 適切なバージョンを見つけ（クラスタに複数ある場合）、[ダウンロード（Download）] をクリックします。

e) RPM パッケージを展開用の Linux ホストにコピーし、ルート権限で次のコマンドを実行します。`rpm -ivh<rpm_filename>`

ステップ 2 **API を使用した新しいトークンの取得**

Secure Connector トークンは、OpenAPI（トークンを取得）を使用して取得することもできます。次の Python および Bash スニペットを使用して、新しいトークンを取得できます。使用される API キーには *external_integration* 機能が必要であり、指定されたルート範囲への書き込みアクセスが必要であることに注意してください。Python 用の Secure Workload OpenAPI クライアントのインストールと新しい API キーの作成については、「OpenAPI 認証」を参照してください。

- **トークン取得のための Python スニペット**

```
from tetpyclient import RestClient
from urllib import quote

API_ENDPOINT = "https://<UI_VIP_OR_DNS_FOR_TETRATION_DASHBOARD>"
ROOT_SCOPE_NAME = r"""<ROOT_SCOPE_NAME>"""
API_CREDENTIALS_FILE = "<API_CREDENTIALS_JSON_FILE>"
OUTPUT_TOKEN_FILE = "registration.token"

if __name__ == "__main__":
  client = RestClient(API_ENDPOINT,
                      credentials_file=API_CREDENTIALS_FILE)  # Add (verify=False)
to skip certificate verification
  escaped_root_scope_name = quote(ROOT_SCOPE_NAME, safe='')
 resp = client.get('/secureconnector/name/{}/token'.format(escaped_root_scope_name))

  if resp.status_code != 200:
    print 'Error ({}): {}'.format(resp.status_code, resp.content)
    exit(1)
  else:
    with open(OUTPUT_TOKEN_FILE, 'w') as f:
      f.write(resp.content)
```

- **トークン取得のための BASH スニペット**

```
#!/bin/bash
HOST="https://<UI_VIP_OR_DNS_FOR_TETRATION_DASHBOARD>"
API_KEY="<API_KEY>"
API_SECRET="<API_SECRET>"
ROOTSCOPE_NAME="<ROOT_SCOPE_NAME>" # if the name contains spaces or special characters,
 it should be url-encoded
TOKEN_FILE="registration.token"
INSECURE=1 # Set to 0 if you want curl to verify the identity of the cluster

METHOD="GET"
URI="/openapi/v1/secureconnector/name/$ROOTSCOPE_NAME/token"
CHK_SUM=""
CONTENT_TYPE=""
TS=$(date -u "+%Y-%m-%dT%H:%M:%S+0000")
CURL_ARGS="-v"
if [ $INSECURE -eq 1 ]; then
    CURL_ARGS=$CURL_ARGS" -k"
fi

MSG=$(echo -n -e "$METHOD\n$URI\n$CHK_SUM\n$CONTENT_TYPE\n$TS\n")
SIG=$(echo "$MSG"| openssl dgst -sha256 -hmac $API_SECRET -binary | openssl enc
-base64)
REQ=$(echo -n "curl $CURL_ARGS $HOST$URI -w '%{http_code}' -H 'Timestamp: $TS' -H
'Id: $API_KEY' -H 'Authorization: $SIG' -o $TOKEN_FILE")
status_code=$(sh -c "$REQ")
if [ $status_code -ne 200 ]; then
```

```
    echo "Failed to get token. Status: " $status_code
else
    echo "Token retrieved successfully"
fi
```

ステップ 3 トークンをコピーしてクライアントを開始する

```
systemctl stop tetration-secure-connectorsystemctl stop tetration-secure-connector
```

トークンファイルを *cert* フォルダにコピーします。

```
/etc/tetration/cert/registration.token
```

Secure Connector クライアントを再起動します。

```
systemctl start tetration-secure-connector
```

Secure Connector クライアントの状態の確認

rpmdb でパッケージ tet-secureconnector-client-site をクエリすることで、Secure Connector クライアントがインストールされているか確認できます。

```
rpm -q tet-secureconnector-client-siterpm -q tet-secureconnector-client-site
```

インストールされているクライアントの現在の状態を確認するには、systemd サービス tetration-secure- connector のステータスを確認します。

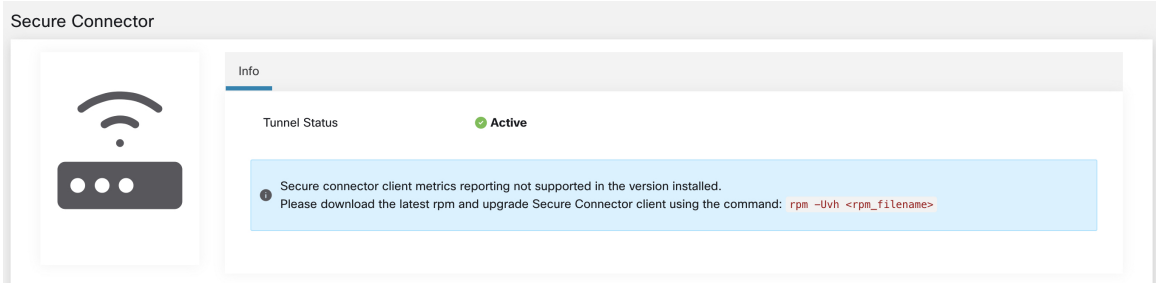
```
systemctl status tetration-secure-connector
```

Secure Connector クライアントステータス

[外部オーケストレータ（External Orchestrators）] ページに、設定された外部オーケストレータと Secure Connector トンネルのステータスが表示されます。外部オーケストレータの設定中に Secure Connector が有効になっている場合、[Secure Connector] ページで Secure Connector クライアントメトリックを表示できます。

ただし、Secure Connector トンネルのステータスが [アクティブ（Active）] であるにもかかわらず、クライアントメトリックが表示されない場合は、古いバージョンの Secure Connector がインストールされていることを意味します。Secure Connector クライアントバージョンのアップグレードに関する、次のようなメッセージが表示されます。

図 *45 : Secure Connector* クライアントのアップグレードメッセージ



（注） 最新の Secure Connector RPM のインストール手順については、「最新の Secure Connector クライアント RPM のダウンロード」を参照してください。

クライアントメトリックを表示するには、次の手順を実行します。

手順

ステップ 1 [設定の詳細（Configure Details）] で、[ステータス（Status）] 行をクリックします。[Secure Connector] ページが表示されます。

（注） 左ペインで**[管理（Manage）] > [Secure Connector]**を選択して、Secure Connector トンネルのステータスにアクセスすることもできます。

ステップ 2 [全般（General）]、[インターフェイス（Interface）]、または[ルート（Routes）] タブを選択して、クライアントと Secure Workload クラスタ間の接続ステータスの詳細にアクセスします。

タブ	説明
General	次の情報を一覧表示します。 • Tunnel Status • ホスト名 • IP アドレス • HTTP/HTTPS プロキシ（HTTP/HTTPS Proxy） • [バージョン（Version）]：ビルドバージョンを一覧表示します。 • vCPU の数 • 合計メモ（GB） • [稼働時間（Uptime）]：Secure Connector クライアントが実行されている VM の稼働時間を一覧表示します。 • [ハートビートの最終受信日時（Last Heartbeat Received）]：クライアントから最後に受信したハートビートの日付とタイムスタンプを一覧表示します。 • [ハートビートの失敗回数（過去 1 日）（No. of Heartbeat Failures (Last 1 day)）]：Secure Connector クライアントへの接続が 1 日に失敗した回数をリストします。クライアントが非アクティブのままである場合、カウントは増えません。カウントは 1 日の終わりにリセットされます。 • ラウンドトリップ遅延（ミリ秒）
インターフェイス	Secure Connector クライアントが実行されている VM のインターフェイスの詳細を一覧表示します。
ルート	ルートテーブルには、宛先 IP アドレス、ゲートウェイ、genmask、およびインターフェイスが一覧表示されます。

Secure Connector クライアントのアップグレード

Secure Connector クライアントは自動更新をサポートしていません。ソフトウェアの新しいバージョンをインストールするには、次のコマンドを使用して現在のバージョンをアンインストールしてから、新しいバージョンの（インストール手順）に進みます。

rpm -e tet-secureconnector-client-siterpm -e tet-secureconnector-client-site

Secure Connector クライアントの削除

Secure Connector クライアントは、次のコマンドを使用してアンインストールできます。

rpm -e tet-secureconnector-client-siterpm -e tet-secureconnector-client-site

Amazon Web Services

（注） AWS 外部オーケストレータ機能は、新しい AWS クラウドコネクタ機能の一部になりました。このリリースにアップグレードすると、既存の AWS 外部オーケストレータは読み取り専用になります。変更が必要な場合は、新しい AWS コネクタを作成します。詳細については、「AWS コネクタ」を参照してください。

Secure Workload では AWS リージョンからリアルタイムでインベントリデータを自動的に取り込むことができます。「AWS」タイプの外部オーケストレータ設定が追加されると、Secure Workload アプライアンスは AWS エンドポイントに接続し、実行中および停止状態のすべてのインスタンスのメタデータを取得します。

前提条件

- 使用されるセキュリティトークン（アクセスキーと秘密鍵）には、オーケストレータ情報の取得を許可する適切な種類の IAM 権限が必要です。

設定フィールド

属性	説明
ID	オーケストレーションの固有識別子。
名前	ユーザーが指定したオーケストレータの名前。
タイプ	オーケストレータのタイプ：（この場合は aws）
説明	オーケストレータの簡単な説明。

属性	説明
AWS アクセス キー ID	オーケストレータ構成が作成されているアカウントに関連付けられているアクセスキー。
AWS シークレットアクセスキー	オーケストレータ構成が作成されているアカウントに関連付けられている秘密鍵。構成を編集するたびに秘密鍵を再入力する必要があることに注意してください。
[AWSリージョン（AWS Region）]	ワークロードが展開されているリージョン。ワークロードが複数のリージョンに分散している場合は、リージョンごとに個別の設定が必要です。正しいリージョン値については、https://docs.aws.amazon.com/general/latest/gr/rande.htmlを参照してください。
自己署名証明書の受け入れ（Accept Self-signed Cert）	AWS の場合、自動的に true とマークされます。ユーザーは編集できません。
フルスナップショット間隔	秒単位のフルスナップショット間隔。Orchestrator Inventory Manager は、オーケストレータからフル更新ポーリングを実行します。
差分スナップショット間隔	秒単位の差分スナップショット間隔。Orchestrator Inventory Manager は、オーケストレータから増分更新のみをフェッチします。
ホストリスト	AWS オーケストレータタイプにはホストリストは必要ありません。AWS のエンドポイントは、前述の AWS リージョンフィールドから取得されます。このフィールドは空白のままにする必要があります。
詳細な TSDB メトリック	有効にすると、個々のオーケストレータの TSDB メトリックがレポートされます。無効な場合、すべてのオーケストレータメトリックの集計がレポートされます。
セキュアコネクタトンネル（Secure Connector Tunnel）	Secure Connector を介したこのオーケストレータのホストへのトンネル接続 します。

ワークフロー

- 前述の構成フィールドを入力した AWS オーケストレータを構成します。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、次のラベルをすべての AWS インスタンスに追加します。

キー	値
orchestrator_system/orch_type	aws
orchestrator_system/cluster_name	<*Cluster_name* はユーザーがこのオーケストレータの設定に付けた名前>
orchestrator_system/cluster_id	<|*product*| でのオーケストレータの設定の *UUID*>

インスタンス固有のラベル

次のラベルはインスタンス固有です。

キー	値
orchestrator_system/workload_type	vm
orchestrator_system/machine_id	<*AWS* によって割り当てられた *InstanceID*>
orchestrator_system/machine_name	<*AWS* によってこのノードに指定された *PublicDNS (FQDN)*>
orchestrator_‘<AWS Tag Key>‘	<*AWS* タグ値>

トラブルシューティング

- AWS リージョンと可用性ゾーンを混同する。

  これらの値は両方とも相互に関連しているため、混同しないでください。たとえば、us-west-1 がリージョンの場合、可用性ゾーンは us-west-1a、us-west-1b などになります。オーケストレータを構成するときは、リージョンを使用する必要があります。すべてのリージョンについては、https://docs.aws.amazon.com/general/latest/gr/rande.html [英語] を参照してください。

- オーケストレータ構成を更新した後の接続およびクレデンシャルの問題。

  ユーザーは、設定が更新されるたびに *AWS* シークレットキーを再送信する必要があります。

Kubernetes/OpenShift

（注） EKS および AKS 外部オーケストレータ機能は、それぞれ新しい AWS および Azure クラウドコネクタ機能の一部になりました。このリリースにアップグレードすると、既存の EKS および AKS 外部オーケストレータは読み取り専用になります。変更が必要な場合は、新しい AWS または Azure コネクタを作成します。詳細については、「Cloud Connector」の関連トピックを参照してください。

シンプルな Kubernetes および OpenShift の外部オーケストレータは変更されていません。

Cisco Secure Workload は、Kubernetes クラスタからライブのインベントリの自動取り込みをサポートします。Kubernetes/OpenShift クラスタに外部オーケストレータ構成が追加されると、Secure Workload はクラスタの API サーバーに接続し、そのクラスタ内のノード、ポッド、およびサービスのステータスを追跡します。Secure Workload は、オブジェクトタイプごとに、オブジェクトに関連付けられているすべての Kubernetes ラベルをインポートします。値はすべてそのままインポートされます。

Secure Workload は、Kubernetes/OpenShift オブジェクト用に定義されたラベルのインポートに加えて、それらのオブジェクトをインベントリフィルタで使用しやすくするラベルも生成します。それらの追加ラベルは、範囲とポリシーを定義する際に特に役立ちます。

すべてのラベルの詳細については、「Kubernetes クラスタに関連するラベル」を参照してください。

Kubernetes ノードで適用が有効になっている場合（適用エージェントがインストールされ、構成プロファイルで適用エージェントでの適用が有効になっている）、適用ポリシーは、この統合を介して取り込まれた Kubernetes エンティティに関する情報を使用して、ノードおよびポッド名前空間の内部の両方にインストールされます。

クラウドプラットフォーム上の Kubernetes について

サポートされているクラウドプラットフォームで実行されている次のマネージド Kubernetes サービスの場合、このオーケストレータの機能は、クラウドコネクタを使用して提供されます。

- Amazon Web Services（AWS）で実行される Elastic Kubernetes Service（EKS）
- Microsoft Azure で実行される Azure Kubernetes Service（AKS）
- Google Cloud Platform（GCP）で実行される Google Kubernetes Engine（GKE）

クラウドプラットフォーム上の Kubernetes クラスタからデータを取得する方法の詳細については、「Cloud Connector」のトピックを参照してください。

要件および前提条件

- サポートされている Kubernetes および OpenShift バージョンについては、https://www.cisco.com/go/secure-workload/requirements/integrationsを参照してください。
- Secure Connector トンネル（接続に必要な場合）。

設定フィールド

次の設定フィールドは、オーケストレータオブジェクトの Kubernetes オーケストレータ設定に関連します。

フィールド	説明
名前	ユーザーが指定したオーケストレーションの名前。
[説明（Description）]	オーケストレーションのユーザー指定の説明。
[デルタ間隔（Delta Interval）]	Kubernetes エンドポイントの変更を確認する間隔（秒）
[フルスナップショット間隔（Full Snapshot Interval）]	Kubernetes データの完全なスナップショットを実行する間隔（秒単位）
[ユーザー名（Username）]	オーケストレーション エンドポイントのユーザー名。
[パスワード（Password）]	オーケストレーション エンドポイントのパスワード。
[証明書（Certificate）]	認証に使用されるクライアント証明書。
[キー（Key）]	クライアント証明書に対応するキー。
[認証トークン（Auth Token）]	Opaque 認証トークン（ベアラートークン）。
[CA証明書（CA Certificate）]	オーケストレーション エンドポイントを検証する CA 証明書。
[自己署名証明書の受け入れ（Accept Self-signed Cert）]	Kubernetes API サーバー証明書の厳密な SSL チェックを無効にするチェックボックス
[詳細なTSDBメトリック（Verbose TSDB Metrics）	Kubernete オーケストレータのメトリックごとに維持されます。False に設定されている場合、Secure Workload クラスタ全体のメトリックのみが維持されます。

フィールド	説明
[セキュアコネクタトンネル（Secureconnector Tunnel）]	このオーケストレータのホストへのセキュアコネクタトンネルを介したトンネル接続
[ホストリスト（Hosts List）]	オーケストレータへの Secure Workload の接続方法を指定する {“host_name”, port_number} ペアの配列
[K8sマネージャータイプ（K8s manager type）]	Kubernetes クラスタのマネージャタイプ（Vanilla/Openshift Kubernetes の環境以外）
[AWSクラスタ名（AWS cluster name）]	クラスタの作成時に指定されたオーケストレータの名前（既存の EKS のみ）
[AWSアクセスID（AWS Access ID）]	オーケストレータ構成が作成されているアカウントに関連付けられているアクセスキー（既存の EKS のみ）
[AWS秘密アクセスキー（AWS Secret Access Key）]	オーケストレータ構成が作成されているアカウントに関連付けられている秘密鍵。構成を編集するたびに秘密鍵を再入力する必要があることに注意してください（既存の EKS のみ）。
[AWSリージョン（AWS Region）]	ワークロードが展開されているリージョン。ワークロードが複数のリージョンに分散している場合は、リージョンごとに個別の設定が必要です。正しいリージョン値については、「https://docs.aws.amazon.com/general/latest/gr/rande.html」を参照してください（既存の EKS のみ）。
[AWS引継ぎロールのARN（AWS Assume Role ARN）]	オーケストレータへの接続中に引き継ぐロールの Amazon リソース番号。「https://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html」を参照してください（既存の EKS のみ）。
[AzureテナントID（Azure Tenant ID）]	Azure サブスクリプションに関連付けられたテナント ID （既存の AKS のみ）。
[AzureクライアントID（Azure Client ID）]	Azure AD で認証する必要があるアプリケーションに関連付けられたグローバルに一意の ID （既存の AKS のみ）。

フィールド	説明
[Azureクライアントシークレット（Azure Client Secret）]	Azure AD で認証する必要があるアプリケーションのサービスプリンシパルに関連付けられたパスワード（既存の AKS のみ）。

オーケストレータのゴールデンルール

ゴールデンルール オブジェクトの属性については、以下を参照してください。ゴールデンルールを使用することで、Kubernetes クラスタノードで適用が有効になった後も、Kubernetes クラスタが機能し続けるために必要なルールの仕様を簡潔に作成できます。

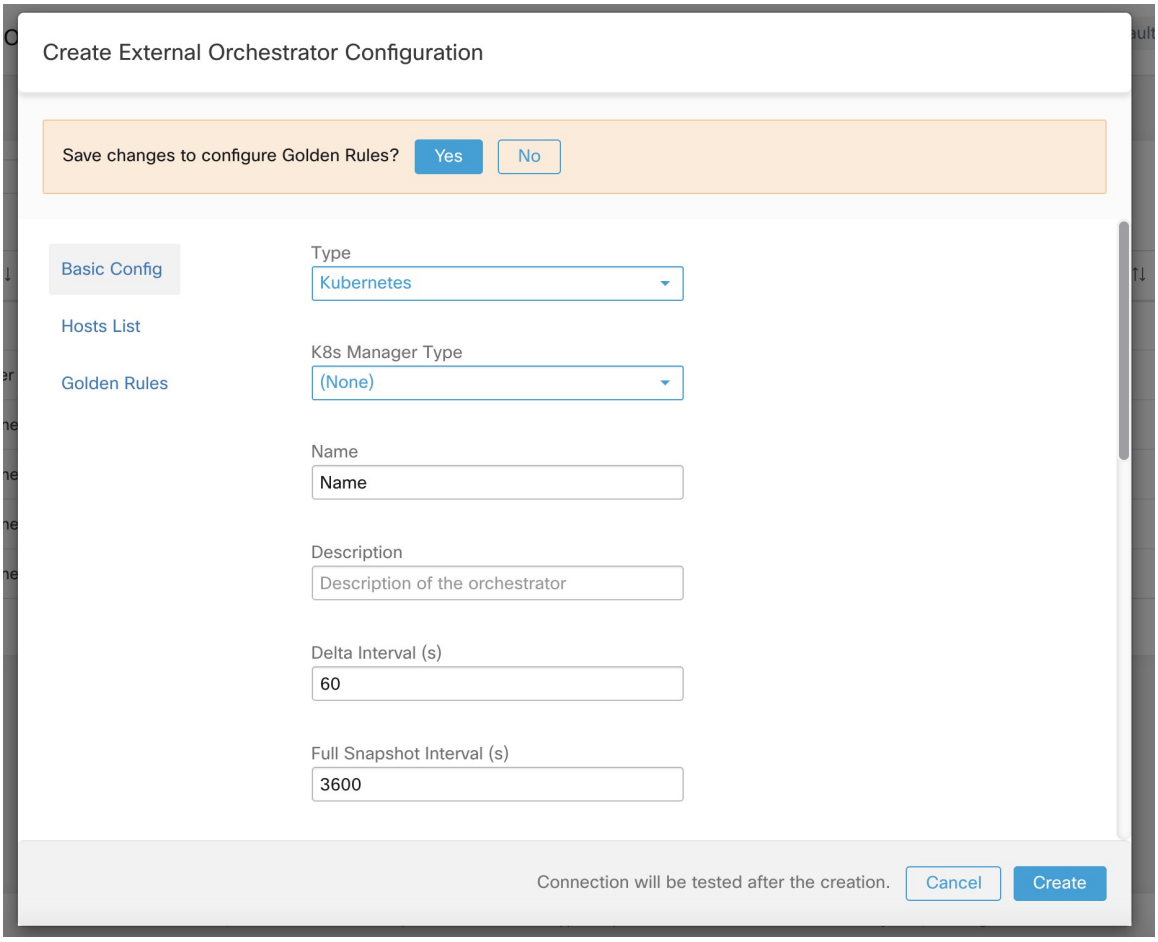
属性	説明
Kubelet ポート	Kubelet ノードのローカル API ポート
サービス	Kubernetes サービスオブジェクトの配列

kubelet ポートは、Kubernetes 管理デーモンから kubelet へのトラフィック（ライブログ、インタラクティブモードのポッドの実行など）を許可するポリシーを作成するために必要です。さまざまなKubernetesサービスとデーモン間の重要な接続は、一連のサービスとして指定され、各サービス配列のエントリには次の構造があります。

- 説明：サービスを説明する文字列
- アドレス：<IP>:<port>/<protocol> 形式のサービス エンドポイント アドレスのリスト。
- コンシューマ：エンドポイントのコンシューマのリスト（許可される値はポッドまたはノード）。

（注） タイプとして [Kubernetes] が選択されている場合、ゴールデンルール構成が許可されます。

図 *46 : Kubernetes* タイプのゴールデンルール構成の作成



ワークフロー

- 必要に応じて、Secure Workload クラスタから Kubernetes API サーバー（複数可）への接続用に Secure Connector トンネルを構成します。
- 前述の構成フィールドを入力した Kubernetes オーケストレータを構成します。
- Kubernetes オーケストレータのゴールデンルールを設定します。

Kubernetes ロールベース アクセス コントロール（RBAC）リソースに関する考慮事項

Kubernetes クライアントは、次のリソースを GET/LIST/WATCH しようとします。管理キー/証明書または管理サービスアカウントを構成しないことを強くお勧めします。

指定する Kubernetes 認証の資格情報には、次のリソースに対する最小限の権限セットが必要です。

リソース	Kubernetes の動詞
endpoints	[get list watch]
名前空間	[get list watch]
ノード	[get list watch]
ポッド	[get list watch]
サービス	[get list watch]
ingresses	[get list watch]
replicationcontrollers	[get list watch]
replicasets	[get list watch]
deployments	[get list watch]
daemonsets	[get list watch]
statefulsets	[get list watch]
雇用	[get list watch]
cronjobs	[get list watch]

基本的に、これらの最小限の権限を使用して、Kubernetes サーバーに特別なサービスアカウントを作成できます。このサービスアカウントの作成を容易にする kubectl コマンドのシーケンスの例を以下に示します。clusterrole（role ではない）と clusterrolebindings（rolebindings ではない）を使用することに注意してください。これらはクラスタ全体のロールであり、名前空間ごとのロールではありません。role/rolebinding の使用は、機能しません。これは、Secure Workload がすべての名前空間からのデータ取得を試行するためです。

```
$ kubectl create serviceaccount csw.read.only
```

clusterrole を作成します。

最小限の権限を持つ clusterrole.yaml の例を以下に示します。

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: csw.read.only
rules:
  - apiGroups:
    - ""
    resources:
      - nodes
      - services
      - endpoints
```

```
      - namespaces
      - pods
      - replicationcontrollers
      - ingresses
    verbs:
      - get
      - list
      - watch
  - apiGroups:
    - extensions
    - networking.k8s.io
    resources:
      - ingresses
    verbs:
      - get
      - list
      - watch
  - apiGroups:
    - apps
    resources:
      - replicasets
      - deployments
      - statefulsets
      - daemonsets
    verbs:
      - get
      - list
      - watch
  - apiGroups:
    - batch
    resources:
      - jobs
      - cronjobs
    verbs:
      - get
      - list
      - watch
```

```
$ kubectl create -f clusterrole.yaml
```



（注） これらのさまざまなリソースの API グループは、Kubernetes のバージョン間の変更の影響を受ける場合があります。上記のサンプルは Kubernetes バージョン 1.20 ～ 1.24 で動作しますが、他のバージョンではいくつかの調整が必要になる場合があります。

clusterrolebinding を作成します。

```
$ kubectl create clusterrolebinding csw.read.only --clusterrole=csw.read.
‹→only --serviceaccount=default:csw.read.only
```

serviceaccount から認証トークンシークレットを取得し（GUI の [認証トークン（Auth Token）] フィールドで使用）、base64 からデコードするために、yaml 出力で serviceaccount をリストして、シークレットの名前を取得できます。

```
$ kubectl get serviceaccount -o yaml csw.read.only
apiVersion: v1
kind: ServiceAccount
metadata:
  creationTimestamp: 2020-xx-xxT19:59:57Z
```

```
  name: csw.read.only
  namespace: default
  resourceVersion: "991"
  selfLink: /api/v1/namespaces/default/serviceaccounts/e2e.minimal
  uid: ce23da52-a11d-11ea-a990-525400d58002
secrets:
- name: csw.read.only-token-vmvmz
```

yaml 出力モードでシークレットをリストすると、トークンが生成されますが、Base64 形式となります（シークレットデータに対する標準の Kubernetes 手順）。Secure Workload は、この形式のトークンを受け入れないため、Base64 からデコードする必要があります。

```
$ kubectl get secret -o yaml csw.read.only-token-vmvmz
apiVersion: v1
data:
  ca.crt: ...
  namespace: ZGVmYXVsdA==
  token: ZXlKaGJHY2lPaUpTVX....HRfZ2JwMVZR
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: csw.read.only
    kubernetes.io/service-account.uid: ce23da52-a11d-11ea-a990-525400d58002
  creationTimestamp: 2020-05-28T19:59:57Z
  name: csw.read.only-token-vmvmz
  namespace: default
  resourceVersion: "990"
  selfLink: /api/v1/namespaces/default/secrets/csw.read.only-token-vmvmz
  uid: ce24f40c-a11d-11ea-a990-525400d58002
type: kubernetes.io/service-account-token
```

シークレットをリストし、`.data.token` フィールドのみを出力し、1 つのコマンドで Base 64 エンコーディングからデコードするには、`--template` オプションを使用する次のコマンドが役立ちます。

```
$ kubectl get secret csw.read.only-token-vmvmz --template "{{ .data.token }}" | base64
-d
```

この認証トークンは、Secure Workload UI で Kubernetes オーケストレータを構成するために、ユーザー名/パスワードまたはキー/証明書の代わりに使用できます。

オーケストレータにより生成されるラベル

「Kubernetes クラスタに関連するラベル」を参照してください。

トラブルシューティング

- クライアントキー/証明書の資格情報の解析/不一致

  これらは PEM 形式で提供し、kubectl.conf ファイルからの正しいエントリである必要があります。お客様が CA 証明書をクライアント証明書フィールドに貼り付けたり、キーと証明書が互いに一致していないことがありました。

- GKE 認証情報の代わりに gcloud 認証情報

gcloud CLI で GKE を使用しているお客様が、GKE クラスタの認証情報が必要な場合に、誤って gcloud 認証情報を提供します。

- Kubernetes クラスタのバージョンがサポートされていません

  互換性のないバージョンの Kubernetes を使用すると、エラーが発生する可能性があります。Kubernetes のバージョンが、サポートされているバージョンのリストにあることを確認してください。

- 資格情報に十分な権限がありません

  使用されている認証トークンや、ユーザーまたはクライアントのキー/証明書に、上記の表にリストされているすべての権限があることを確認します。

- Kubernetes インベントリが変動し続けています

  hosts_list フィールドは、同じ Kubernetes クラスタの API サーバーのプールを指定します。これを使用して複数の Kubernetes クラスタを設定することはできません。Secure Workload は稼働状態を調べ、これらのエンドポイントの 1 つをランダムに選択して接続し、Kubernetes インベントリ情報を取得します。ここではロードバランシングは実行されず、これらのエンドポイント間で負荷が均等に分散される保証もありません。これらが異なるクラスタである場合、どのクラスタの API サーバーに接続するかに応じて、Kubernetes インベントリはそれらのクラスタ間を切り替え続けます。

- 複数の認証方式

  複数の認証方式（ユーザー名/パスワード、認証トークン、クライアントキー/証明書）が設定中に入力される場合があり、API サーバーとの間で確立されたクライアント接続で使用されます。ここでは、有効な同時認証方式に対する標準の Kubernetes ルールが適用されます。

- SSL 証明書の検証失敗

  Kubernetes API エンドポイントが NAT またはロードバランサの背後にある場合、kube コントロールプレーンノードで生成された SSL 証明書の DN が、Cisco Secure Workload で設定された IP アドレスと一致しない場合があります。これにより、CA 証明書が提供され、有効であっても、SSL 検証が失敗します。Insecure ノブは厳密なサーバー SSL 証明書の検証をバイパスするため、この問題を回避するのに役立ちますが、MITM の問題につながる可能性があります。これに対する正しい修正は、CA 証明書を変更して、Kubernetes クラスタへの接続に使用できるすべての DNS/IP エントリに SAN（サブジェクト代替名）エントリを指定することです。

VMware vCenter

vCenter 統合により、ユーザーは設定された vCenter からベアメタルおよび VM 属性を取得できます。

「vCenter」タイプに外部オーケストレータ設定が追加されると、Secure Workload はその vCenter インスタンスによって制御されるすべてのベアメタルと VM について、ベアメタル属性と VM

属性を取得します。Secure Workload はベアメタル/VM の次の属性をインポートします。- a) ホスト名 b) IP アドレス c) BIOS UUID d) カテゴリ/ラベル。

インベントリがアプライアンスに存在しない場合、新しいインベントリが上記のベアメタル/VM 属性を使用して Secure Workload で作成されます。インベントリがアプライアンスに既に存在する場合（ベアメタル/VM で実行されている Secure Workload 可視性センサーによって作成）、既存のインベントリは、取得されたベアメタル/VM のカテゴリ/ラベルリストでラベル付けされます。

前提条件

- 接続に必要な場合、Secure Connector トンネル。
- サポートされている vCenter バージョンが 6.5 以降であること

設定フィールド

「外部オーケストレータの作成」で説明されている共通の設定フィールドのほかに、次のフィールドを設定できます。

- [ホストリスト（Hosts List）] は、ベアメタル/VM 属性がフェッチされる vCenter サーバーを指すホスト名/IP とポートのペアの配列です。

ワークフロー

- ユーザーは Secure Workload クラスタから該当 IP やポートを使用して vCenter サーバーにアクセスできることを最初に確認する必要があります。
- TaaS の場合、または vCenter サーバーに直接アクセスできない場合、ユーザーはセキュアなコネクタトンネルを確立して接続を提供する必要があります。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、vCenter サーバーから学習したすべての VM に次のラベルを追加します。

キー	値
orchestrator_system/orch_type	vCenter
orchestrator_system/cluster_name	*<Name given to this cluster's configuration>*
orchestrator_system/cluster_id	*<UUID of the cluster's configuration in |product|>*

インスタンス固有のラベル

次のラベルはインスタンス固有です。

表 *18 :* 次のラベルはインスタンス固有です。

キー	値
orchestrator_system/workload_type	vm
orchestrator_system/machine_id	ベアメタル/VM の BIOS UUID
orchestrator_system/machine_name	ベアメタル/VM のホスト名
orchestrator_‘<Category Name>‘	<Tag Value>

警告

- vCenter 用の外部オーケストレータ構成が追加されると、Secure Workload ソフトウェアはホストリストで指定された vCenter サーバーに接続します。サーバーへの接続が成功すると、Secure Workload ソフトウェアは、vCenter サーバーに存在するすべてのベアメタルと仮想マシンのホスト名、IP アドレス、およびカテゴリ/ラベルをインポートします。ベアメタルと VM のホスト名と IP アドレスをインポートするには、すべてのベアメタルと VM に VM ツールをインストールする必要があります。特定のベアメタル/仮想マシンに VM ツールがインストールされていない場合、Secure Workload ソフトウェアはその特定のベアメタル/VM のカテゴリ/ラベルを表示しません。
- Cisco Secure Workload ソフトウェアは、ベアメタル/VM のカスタム属性をインポートしません。
- vCenter サーバーの負荷を軽減するために、[デルタ（Delta）] インターバルタイマーを 10 分以上に設定することをお勧めします。上記のタイマーが変更されると、vCenter サーバー上のインベントリ/ラベルに変更がある場合に、少なくとも 10 分の伝播遅延が発生します。

トラブルシューティング

- 接続の問題

  Secure Workload アプライアンスが vCenter サーバーに接続または到達できない場合、外部オーケストレータの [接続ステータス（Connection Status）] タブに、適切なエラー（エラーがある場合）とともに障害ステータスが表示されます。

- Cisco Secure Workload ソフトウェアのヘルスチェック。

  [メンテナンス/サービスステータス（MAINTENANCE/Service Status）] ページをチェックして、サービスが停止していないか確認してください。OrchestratorInventoryManager が稼働しているか確認してください。

DNS

DNS 統合により、Secure Workload は、CNAME および A/AAAA レコードからのホスト名などの DNS 情報で、既知のインベントリに注釈を付けることができます。

「dns」タイプの外部オーケストレータ構成が追加されると、Secure Workload アプライアンスは DNS サーバーへの接続を試み、DNS レコードのゾーン転送ダウンロードを実行します。これらのレコード（A/AAAA および CNAME レコードのみ）は解析され、「orchestrator_system/dns_name」と呼ばれる単一の複数値ラベルを使用して、（オーケストレータが設定されているテナントに属する）Secure Workload パイプラインのインベントリを強化するために使用されます。値は、その IP アドレスを（直接的または間接的に）指す DNS エントリになります。

前提条件

- セキュアコネクタトンネル（接続に必要な場合）
- サポートされている DNS サーバー：BIND9、AXFR（RFC 5936）をサポートするサーバー、Microsoft Windows Server 2016

設定フィールド

- [DNSゾーン（DNS zones）] は文字列配列であり、各文字列は DNS サーバーから転送される DNS ゾーンを表します。すべての DNS ゾーンには、末尾にピリオド（「.」）の文字が必要です。
- [ホストリスト（Hosts List）] はホスト名/IP とポートのペアの配列であり、DNS レコードを取得する DNS サーバーを指します。ここでは、HA のみを目的として複数の DNS サーバーを構成できます。hosts_list で複数の DNS サーバーが指定された場合、高可用性では「最初の正常なサーバー」が使用されます。hosts_list のエントリの先頭から順に優先されます。ゾーンを DNS サーバー間で分割することはできません。

ワークフロー

- まず、ユーザーは、Secure Workload クラスタから該当 IP/ポートで DNS サーバーに到達可能なことを確認する必要があります。
- TaaS の場合、または DNS サーバーに直接到達できない場合、ユーザーは安全なコネクタトンネルを設定して接続を提供する必要があります。
- DNS サーバーで正しい DNS ゾーン転送 ACL または構成を設定します。詳細については、個別の DNS サーバーソフトウェアのマニュアルを参照してください。

生成されるラベル

orchestrator_system/dns_name -> その値がすべてその IP を示す CNAME および A/AAAA ホスト名である複数の値フィールド。

警告

- DNS オーケストレータフィードはメタデータフィードです。DNS ゾーン転送から学習された IP アドレスによって Cisco Secure Workload にインベントリアイテムが作成されることはありません。代わりに、既存の IP アドレスのラベルが新しい DNS メタデータで更新されます。不明な IP の DNS データは、警告なしで破棄されます。センサー統合または他のオーケストレータ統合を介して学習されていない IP の DNS メタデータに注釈を付けるには、CMDB 一括アップロードメカニズムを使用して IP をアップロードし、IP のインベントリエントリを作成する必要があります。CMDB アップロードから学習されたサブネットによって、インベントリエントリが作成されることはありません。
- DNS サーバーからの CNAME レコードと A/AAAA レコードのみが処理されます。CNAME レコードは、それらが指す A/AAAA レコードを介して最終的な IPv4/IPv6 レコードに処理されます。CNAME が同じオーケストレータからの A/AAAA レコードを指している限り、単一レベルの参照のみがサポートされます（つまり、CNAME -> CNAME -> A/AAAA やそれ以上のチェーンは参照されません）。異なる DNS オーケストレータ間での CNAME 参照はサポートされていません。

トラブルシューティング

- 接続の問題

  Cisco Secure Workload は、Secure Workload アプライアンスサーバーの 1 つから、または TaaS の場合はクラウドから、または Secure Workload Secure Connector VPN トンネルサービスをホストしている VM から発信される TCP 接続を使用して、指定された IP/ホスト名とポート番号への接続を試みます。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。

- DNS AXFR 特権の問題

  さらに、ほとんどの DNS サーバー（BIND9 または Windows DNS または Infoblox）では、クライアント IP が DNS ゾーン転送（DNS プロトコルのオペコードによる AXFR 要求）を試みるときに、追加の設定が必要になります。これは、個々の DNS レコードを解決するためのシンプルな DNS 要求と比べて、必要なリソース消費量と特権がより多くなるためです。これらのエラーは、通常、AXFR が理由コード 5（REFUSE）で拒否されたときに表示されます。

  したがって、DNS サーバーが正しく設定されていることを確認するための手動テストは、ホスト名ルックアップが成功したかによるのではなく、（dig などのツールを使用して）具体的な AXFR 要求をテストする必要があります。

DNS サーバーからの AXFR ゾーン転送の実行に失敗すると、Secure Workload アプライアンスによって「authentication_failure_error」フィールドで報告されます。

また、Secure Workload は、設定されたすべての DNS ゾーンからのゾーン転送を試みますが、DNS データを Secure Workload ラベルデータベースに挿入するためには、そのすべてが成功する必要があることに注意してください。

- インベントリの [ホスト名（Hostname）] フィールドは [DNS] フィールドによって入力されず、「hostname」は常に Secure Workload センサーから学習されます。インベントリがセンサーからではなく、CMDB アップロードを介してアップロードされた場合、ホスト名が欠落している場合があります。DNS オーケストレータ ワークフローからのすべてのデータは、「orchestrator_system/dns_name」ラベルの下にのみ表示され、ホスト名フィールドに入力されることはありません。

DNS オーケストレータのフル/差分ポーリングの動作

デフォルトのフルスナップショット間隔は 24 時間です

デフォルトの差分スナップショット間隔は 60 分です

デフォルトの間隔は、各タイマーの最小許容値でもあります。

DNS レコードはほとんど変更されないため、最適なフェッチ動作のために、Secure Workload は差分スナップショット間隔ごとに、いずれかの DNS ゾーンのシリアル番号が前の間隔から変更されているかチェックします。ゾーンが変更されていない場合、アクションは必要ありません。

いずれかのゾーンが変更された場合は、構成されているすべての DNS ゾーン（変更された単一のゾーンだけでなく）からゾーン転送を実行します。

Secure Workload は、完全なスナップショット間隔ごとに、ゾーンのシリアル番号が変更されたかどうかに関係なく、すべてのゾーンからゾーン転送ダウンロードを実行して、ラベルデータベースに挿入します。

サポートされない機能

警告

- DNAME エイリアスとルックアップはサポートされていません。
- 増分ゾーン転送 (IXFR) はサポートされていません。

Infoblox

Infoblox 統合により、Secure Workload は Infoblox サブネット、ホスト（record:host）、および A/AAAA レコードを Secure Workload インベントリデータベースにインポートできるようにな

ります。拡張可能な属性の名前と値はそのままインポートされ、範囲と適用ポリシーを定義するための Secure Workload ラベルとして使用できます。

（注） 拡張可能な属性を持つ Infoblox オブジェクトのみが考慮されます。すなわち、拡張可能な属性がアタッチされていないものは、インポートから除外されます。

下の図は、 Infoblox からインポートされた、拡張可能な属性 Department を持つホストオブジェクトに対して生成されたラベルの例を示しています。

図 *47 : Infoblox* ラベルの例

```
1. orchestrator_Department = AES789
2. orchestrator_system/cluster_id = [redacted]
3. orchestrator_system/cluster_name = scale13-ib
4. orchestrator_system/machine_id =
   record:host/[redacted]:client8/%20
5. orchestrator_system/machine_name = client8
6. orchestrator_system/orch_type = infoblox
```

前提条件

- WAPI バージョン 2.6、2.6.1、2.7、2.7.1 をサポートする Infoblox REST API エンドポイント（推奨）

設定フィールド

「外部オーケストレータの作成」で説明されている共通の設定フィールドのほかに、次のフィールドを設定できます。

共通フィールド	必須	説明
[ホストリスト（Hosts List）]	○	ホストリストは、1 つの Infoblox グリッドを示します。つまり、REST API アクセス権を持つ複数のグリッドメンバーを追加できます。接続エラーが発生した場合、外部オーケストレータはリスト内の次のメンバーに切り替えます。別の Infoblox グリッドからラベルをインポートする場合は、新しい外部オーケストレータを作成してください。

（注） Infoblox 外部オーケストレータの場合、IPv4 および IPv6（デュアルスタックモード）アドレスがサポートされます。ただし、デュアルスタックのサポートはベータ版の機能であることに注意してください。

ワークフロー

- まず、ユーザーは、Secure Workload クラスタから Infoblox REST API エンドポイントに到達できることを確認する必要があります。
- TaaS の場合、または Infoblox サーバーに直接到達できない場合、ユーザーは Secure Connector トンネルを設定して接続を可能にする必要があります。
- *Infoblox* タイプの外部オーケストレータを作成します。Infoblox データの量（サブネット、ホスト、および A/AAAA レコードの数）によっては、最初の完全なスナップショットが Cisco Secure Workload で使用可能になるまでに最大 1 時間かかります。
- Infoblox 設定の際に、ユーザーはいずれかのレコードタイプ（サブネット、ホスト、A/AAAA レコード）の選択を解除することもできます。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、Infoblox から取得したすべてのオブジェクトに次のシステムラベルを追加します。

キー	値
orchestrator_system/orch_type	InfoBlox
orchestrator_system/cluster_id	<UUID of the external orchestrator in Secure Workload
orchestrator_system/cluster_name	<外部オーケストレータに割り当てられた名前>
orchestrator_system/machine_id	<Infoblox object reference/identifier>
Orchestrator_system/machine_name	<Infoblox host (DNS) name>

生成されるラベル

すべての Infoblox 拡張可能属性は、プレフィックス *orchestrator_* の付いた Secure Workload ラベルとしてインポートされます。たとえば、*Department* という拡張可能属性を持つホストは、Secure Workload のインベントリ検索で *orchestrator_Department* としてアドレス指定できます。

キー	値
orchestrator_<extensible attribute>	<value(s) of the extensible attribute as retrieved from Infoblox>

警告

- Infoblox からインポートできるサブネットの最大数は 50000 です。
- Infoblox からインポートできるホストと A/AAAA レコードの最大数は、合計で 400000 です。

トラブルシューティング

- 接続の問題：Secure Workload は、Secure Workload アプライアンスサーバーの 1 つから、または TaaS の場合はクラウドから、または Secure Workload セキュア コネクタ トンネル サービスをホストしている VM から、HTTPS 接続を使用して、指定された IP/ホスト名とポート番号に接続しようとします。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。また、指定されたクレデンシャルが正しいこと、および REST API 要求を Infoblox に送信するための権限があることを確認してください。
- すべての予期されるオブジェクトがインポートされない：Secure Workload ではサブネット、ホスト、および拡張可能な属性が添付された A/AAAA レコードのみがインポートされます。Infoblox からインポートできるオブジェクトの数には制限があることに注意してください（「注意事項」を参照）。
- インベントリでサブネットが見つからない：Secure Workload インベントリは設計上 IP アドレスのみが含まれているため（ホストや A/AAAA レコードなど）、インベントリ検索を使用して Infoblox サブネットを見つけることはできません。
- ホストまたは A/AAAA レコードが見つからない：Secure Workload は Infoblox から取得したすべての拡張可能属性をインポートします。たとえば、インベントリ検索でプレフィックス *orchestrator_* を拡張可能な属性名に追加してください。サブネットの拡張可能属性は、Infoblox で継承とマークされていない場合、ホストの一部ではないため、Cisco Secure Workload で検索できないことに注意してください。

F5 BIG-IP

F5 BIG-IP 統合により、Secure Workload は、F5 BIG-IP ロード バランサ アプライアンスから仮想サーバーをインポートし、サービスインベントリを取得できます。サービスインベントリは F5 BIG-IP 仮想サーバーに対応し、そのサービスは VIP（仮想 IP アドレス）、プロトコル、およびポートによって特性が決まります。Secure Workload にインポートされると、このサービス

インベントリにはservice_nameなどのラベルが付けられます。これは、インベントリ検索で使用できるだけでなく、Secure Workloadの範囲とポリシーを作成するためにも使用できます。

この機能の大きな利点は、F5 BIG-IPの外部オーケストレータがSecure Workloadポリシーを仮想サーバーに割り当てられたセキュリティルールに変換し、REST APIを介してF5 BIG-IPロードバランサに展開するという仕方でポリシーを適用できることです。

前提条件

- セキュアコネクタトンネル（接続に必要な場合）
- F5 BIG-IP REST API エンドポイントバージョン 12.1.1

設定フィールド

「外部オーケストレータの作成」で説明されている共通の設定フィールドのほかに、次のフィールドを設定できます。

フィールド	必須	説明
[ホストリスト（Hosts List）]	○	このフィールドでは、F5 BIG-IPロードバランサのREST API エンドポイントを指定します。F5 BIG-IP用に高可用性が設定されている場合は、他のメンバーノードも入力してください。現在のノードとの通信に失敗した場合、外部オーケストレータによって切り替えられます。別のF5 BIG-IP ロードバランサからラベルをインポートする場合は、新しい外部オーケストレータを作成する必要があります。

フィールド	必須	説明
[適用の有効化（Enable Enforcement）]	×	デフォルト値は false（チェックボックスがオフ）です。チェックすると、Secure Workload のポリシーの適用が許可され、対応する F5 BIG-IP ロードバランサにセキュリティポリシールールを展開できます。指定されたログイン情報には、F5 BIG-IP REST API の書き込みアクセス権が必要であることに注意してください。
ルートドメイン	×	デフォルト値は 0（ゼロ）です。ルートドメインは、外部オーケストレータによって考慮される仮想サーバーを指定します。仮想サーバーは、特定のルートドメインに割り当てられたパーティションのリストによって決定され、それらのパーティションで定義された仮想サーバーのみが Cisco Secure Workload にインポートされます。

ワークフロー

- まず、ユーザーは、Cisco Secure Workload から F5 BIG-IP REST API エンドポイントに到達できることを確認する必要があります。
- TaaS の場合、または F5 BIG-IP アプライアンスに直接到達できない場合、ユーザーは Secure Connector トンネルを設定して接続を可能にする必要があります。
- *F5 BIG-IP* タイプの外部オーケストレータを作成します。
- 差分間隔の値によっては、F5 BIG-IP 仮想サーバーの最初の完全スナップショットが完了するまでに最大 60 秒（デフォルトの差分間隔）かかる場合があります。その後、生成されたラベルを使用して、Secure Workload の範囲と適用ポリシーを作成できます。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、*F5 BIG-IP* の外部オーケストレータに次のシステムラベルを追加します。

キー	値
orchestrator_system/orch_type	f5
orchestrator_system/cluster_id	<外部オーケストレータの UUID>
orchestrator_system/cluster_name	<外部オーケストレータに割り当てられた名前>
orchestrator_system/workload_type	サービス
orchestrator_system/namespace	<仮想サーバーが属するパーティション>
orchestrator_system/service_name	<F5 BIG-IP 仮想サーバー名>

生成されるラベル

外部オーケストレータは、仮想サーバーごとに次のラベルを生成します。

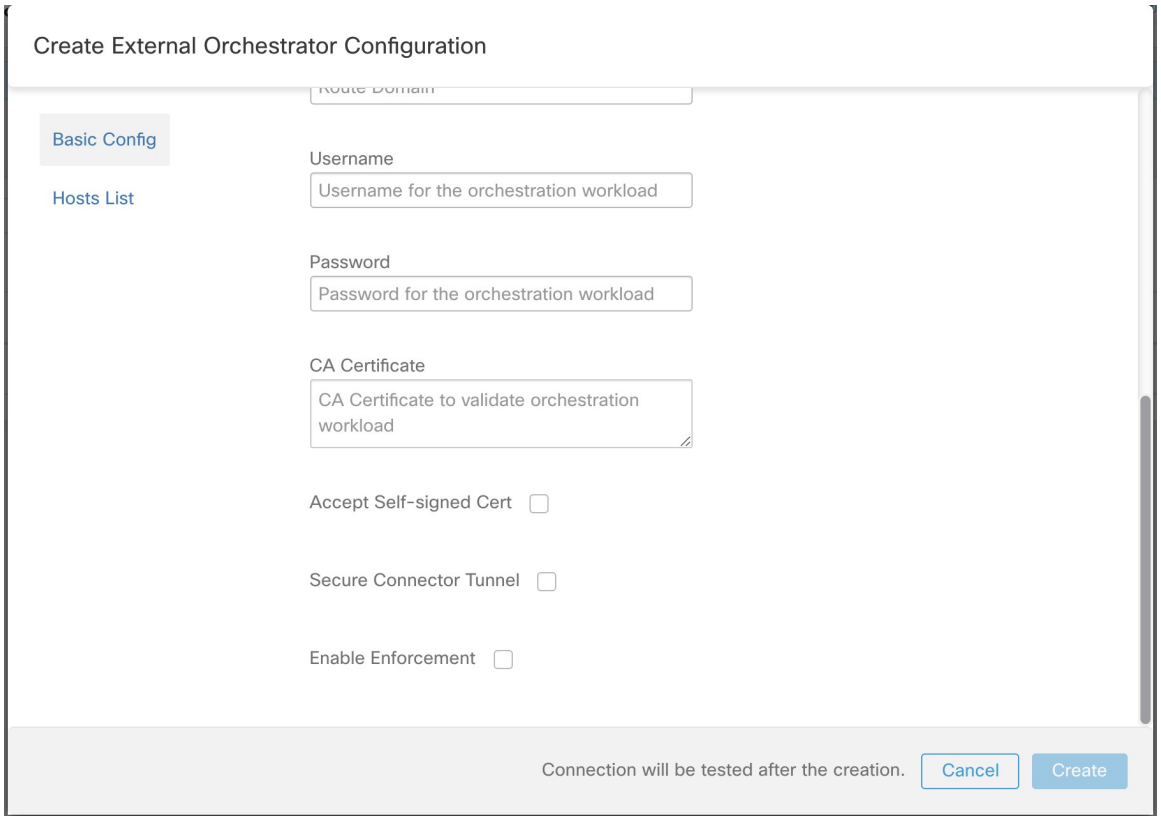
キー	値
orchestrator_annotation/snat_address	<仮想サーバーの SNAT アドレス>

F5 BIG-IP のポリシーの適用

この機能により、Secure Workload は、ラベル付きの F5 BIG-IP 仮想サーバーに一致するプロバイダーグループの論理ポリシーを F5 BIG-IP セキュリティポリシールールに変換し、それらのルールを REST API を使用してロード バランサ アプライアンスに展開できます。前述のように、それぞれの F5 BIG-IP 仮想サーバーへの既存のセキュリティポリシーの割り当ては、Secure Workload によって生成されたセキュリティポリシーを指す新しい割り当てに置き換えられます。既存のセキュリティポリシーが変更されたり、F5 BIG-IP ポリシーリストから削除されたりすることはありません。

デフォルトでは、適用は外部オーケストレータ設定では有効になっていません。

図 *48* : 設定オプション *[*適用の有効化（***Enable Enforcement***）*]*

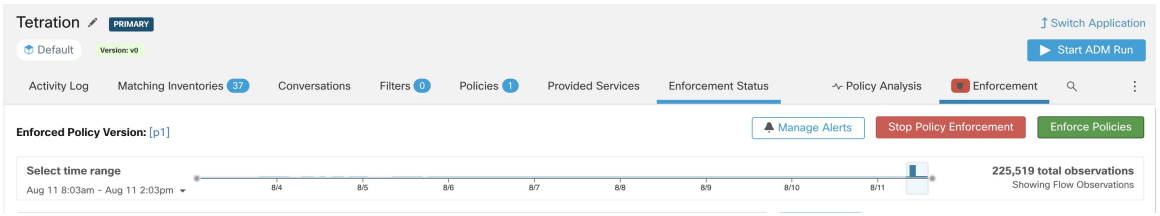


このオプションは、必要に応じていつでも変更できます。

適用を有効にしても、ロードバランサに適用される少なくとも 1 つのポリシーを含むワークスペースで適用を有効にするまで、ロード バランサ アプライアンスにポリシーは展開されません。または、インベントリの更新に起因する場合もあります。

ただし、オーケストレータの適用を無効にすると、展開されたすべてのセキュリティポリシールールが F5 BIG-IP ロードバランサからすぐに削除されます。

図 *49* : ワークスペースポリシーの適用



（注）
- F5 BIG-IP のオーケストレータも、セキュリティポリシールールの逸脱を検出し、Secure Workloadポリシーに置き換えます。そのため、仮想サーバーに対するポリシーの変更は、すべて Secure Workload を使用して行う必要があります。
- ポリシーの適用が停止されるか、外部オーケストレータが削除されると、すべての Secure Workload ポリシーが F5 BIG-IP ロードバランサから削除されるため、仮想サーバーのセキュリティポリシーは空になります。

外部オーケストレータの OpenAPI ポリシー適用ステータスを使用して、外部オーケストレータに関連付けられたロード バランサ アプライアンスへの Secure Workload ポリシー適用のステータスを取得できます。この機能は、F5 BIG-IP アプライアンスへのセキュリティポリシールールの展開が成功したかどうかを確認するのに役立ちます。

F5 入力コントローラのポリシーの適用

Cisco Secure Workload は、ポッドが Kubernetes 入力オブジェクトを使用して外部クライアントに公開されるときに、*F5 BIG-IP* ロードバランサとバックエンドポッドの両方でポリシーを適用します。

F5 入力コントローラを使用したポリシー適用手順は次のとおりです。

手順

ステップ 1 前述したように、*F5 BIG-IP* ロードバランサの外部オーケストレータを作成します。

ステップ 2 ここに記載する説明に従って、Kubernetes または OpenShift の外部オーケストレータを作成します。

```
→  ~
→  ~ k8s get ingress
NAME           HOSTS   ADDRESS          PORTS   AGE
test-ingress   *       192.168.60.100   80      7s
→  ~
```

ステップ 3 Kubernetes クラスタに入力オブジェクトを作成します。入力オブジェクトの作成で使用する yaml ファイルのスナップショットを次の図に示します。

```
→  ~
→  ~ k8s get ingress test-ingress -o yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    virtual-server.f5.com/ip: 192.168.60.100
    virtual-server.f5.com/partition: k8scluster
  creationTimestamp: "2019-07-26T18:34:39Z"
  generation: 1
  name: test-ingress
  namespace: default
  resourceVersion: "8310"
  selfLink: /apis/extensions/v1beta1/namespaces/default/ingresses/test-ingress
  uid: 06f8a705-afd4-11e9-97fb-525400d58002
spec:
  backend:
    serviceName: nginx
    servicePort: 80
status:
  loadBalancer:
    ingress:
    - ip: 192.168.60.100
→  ~
```

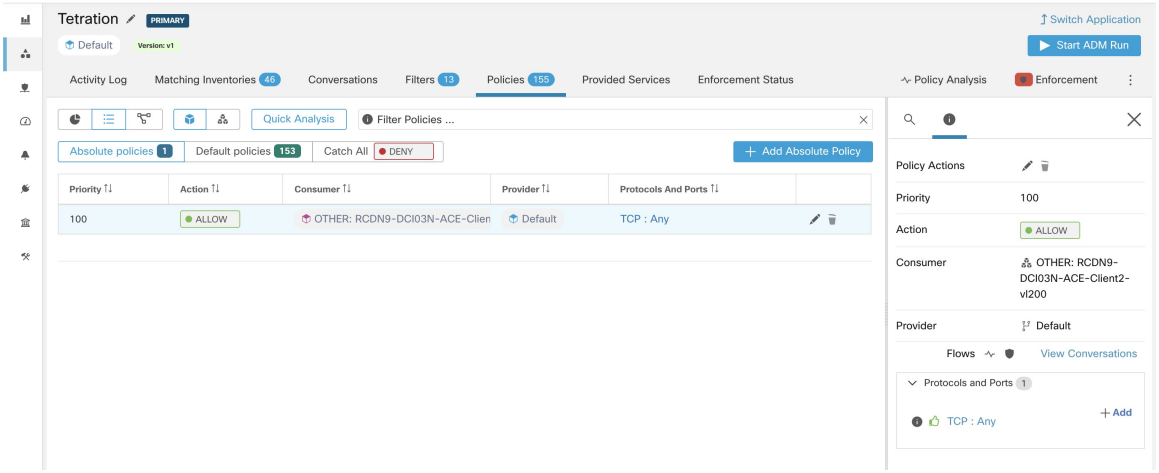
ステップ 4 F5 入力コントローラポッドを Kubernetes クラスタに展開します。

```
→  ~ k8s get deploy -n kube-system
NAME                   DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
coredns                2         2         2            2           31m
k8s-bigip-ctlr-cluster 1         1         1            1           5m20s
→  ~
```

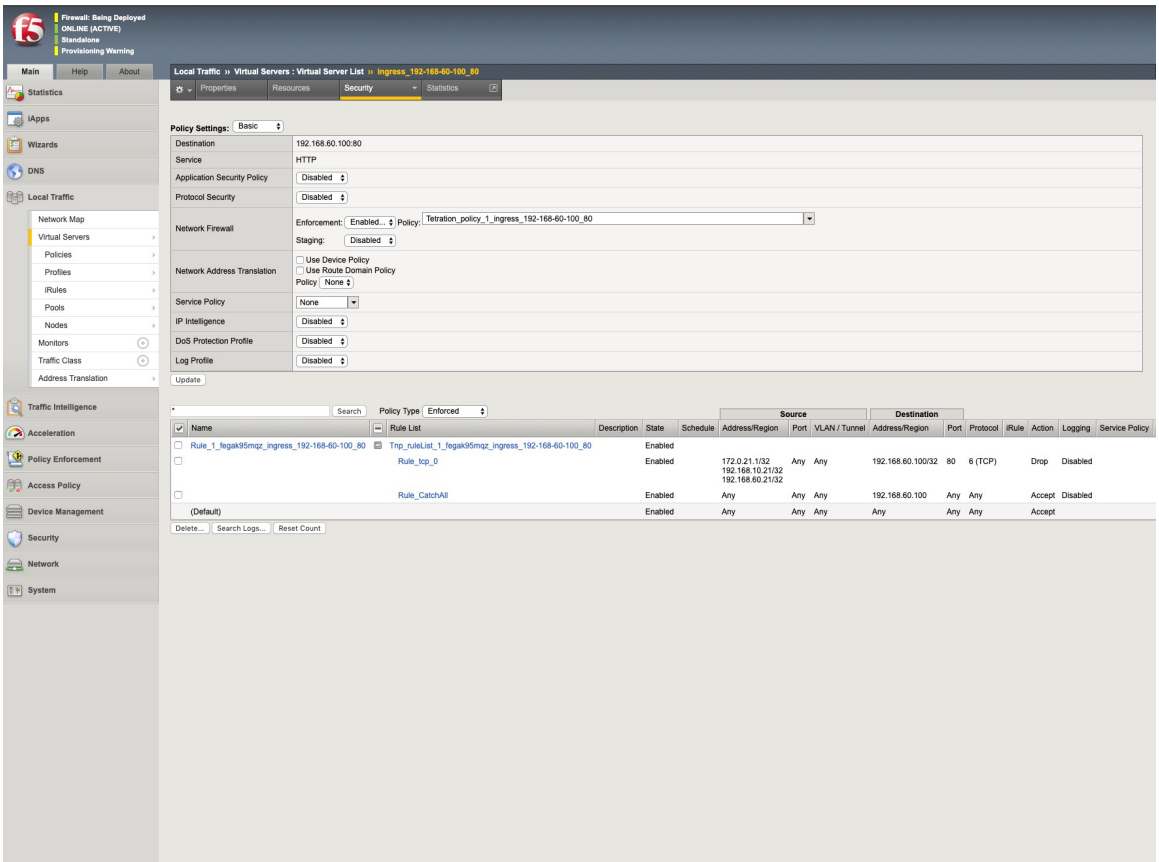
ステップ 5 クラスタ外のコンシューマがアクセスするバックエンド サービスを作成します。以下の例では、*nginx* サービスを作成しています。

```
→  ~
→  ~ k8s get deploy
NAME    DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx   1         1         1            0           5s
→  ~
```

ステップ 6 外部コンシューマとバックエンドサービス間にポリシーを作成します。[ポリシー適用（Policy Enforcement）] タブを使用してポリシーを適用します。



ステップ 7 *F5 BIG-IP* ロードバランサとバックエンドポッドのポリシーを確認します。F5 ロードバランサの場合、Secure Workload は適切な許可またはドロップルールを適用します。送信元はステップ 6 で指定したコンシューマであり、宛先は VIP（F5 の入力仮想サービスの VIP）になります。バックエンドポッドの場合、Secure Workload は適切な許可またはドロップルールを適用します。このとき、送信元が SNIP（SNAT プールが有効になっている場合）または F5 IP（自動マップが有効になっている場合）になり、宛先がバックエンドポッド IP になります。



警告

- F5 BIG-IP HA モードの展開フェーズ中に、構成同期オプションを有効にしてください。有効にすると、外部オーケストレータが、現在接続しているホストから仮想サーバーの最新リストをフェッチできます。
- F5 BIG-IP HA 展開 モードの場合、SNAT プールの代わりに Auto-Map がアドレス変換用に構成されている場合は、プライマリ BIG-IP がフローティングセルフ IP アドレスで構成されていることを確認してください。
- 単一のアドレスとして指定された VIP のみがサポートされています。つまり、サブネットとして指定された VIP はサポートされていません。

トラブルシューティング

- 接続の問題 Secure Workload は、Secure Workload アプライアンスサーバーの 1 つから、または TaaS の場合はクラウドから、または Secure Workload Secure Connector トンネル サービスをホストしている VM から、HTTPS 接続を使用して、指定された IP/ホスト名とポート番号に接続しようとします。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。また、指定されたクレデンシャルが正しいこと、および F5 BIG-IP アプライアンスに REST API 要求を送信するための読み取りおよび書き込みアクセス権限があることを確認してください。
- セキュリティルールが見つからない定義された仮想サーバーのセキュリティルールが見つからない場合は、ポリシー適用の実行後に、対応する仮想サーバーが有効になっていることを確認してください。その可用性/ステータスが使用可能/有効になっている必要があります。

Citrix Netscaler

Citrix Netscaler 統合により、Secure Workload は、Netscaler ロード バランサ アプライアンスからロードバランシング仮想サーバーをインポートし、サービスインベントリを取得できます。サービスインベントリは、仮想サーバーによって提供される Netscaler サービスに対応し、*service_name* などのラベルを指定されます。これらのラベルは、インベントリ検索で使用することが可能で、Secure Workload の範囲とポリシーの作成に使用できます。

この機能の大きな利点は、*Citrix Netscaler* の外部オーケストレータが Secure Workload ポリシーを Netscaler ACL ルールに変換し、REST API を介して Netscaler ロードバランサに展開するという仕方でポリシーを適用できることです。

前提条件

- セキュアコネクタトンネル（接続に必要な場合）
- Netscaler REST API エンドポイントバージョン 12.0.57.19

設定フィールド

「外部オーケストレータの作成」で説明されている共通の設定フィールドのほかに、次のフィールドを設定できます。

共通フィールド	必須	説明
[ホストリスト（Hosts List）]	○	これは、Citrix Netscaler ロードバランサの REST API エンドポイントを指します。高可用性構成の場合は、他のメンバーノードも入力してください。外部オーケストレータは、現行ノードとの通信に失敗した場合に切り替えます。別の Citrix Netscaler ロードバランサからラベルをインポートする場合は、新しい外部オーケストレータを作成する必要があります。
[適用の有効化（Enable Enforcement）]	×	デフォルト値は false（チェックボックスがオフ）です。チェックボックスをオンにすると、Secure Workload のポリシー適用によって、対応する Citrix Netscaler ロードバランサに ACL ルールを展開できます。 指定した資格情報には、Citrix Netscaler REST API への書き込みアクセス権が必要であることに注意してください。

ワークフロー

- まず、ユーザーは、Secure Workload クラスタから Netscaler REST API エンドポイントに到達できることを確認する必要があります。
- TaaS の場合、または Netscaler アプライアンスに直接到達できない場合、ユーザーは Secure Connector トンネルを設定して接続を提供する必要があります。
- タイプが Citrix Netscaler の外部オーケストレータを作成します。

- デルタ間隔の値によっては、Netscaler仮想サーバーの最初の完全スナップショットが完了するまでに最大 60 秒（デフォルトのデルタ間隔）かかる場合があります。その後、生成されたラベルを使用して、Secure Workload の範囲と適用ポリシーを作成できます。
- Secure Workload からポリシーを適用して、Netscaler ACL ルールを展開します。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、Citrix NetScaler の外部オーケストレータに次のシステムラベルを追加します。

キー	値
orchestrator_system/orch_type	nsbalancer
orchestrator_system/cluster_id	<外部オーケストレータの UUID>
orchestrator_system/cluster_name	<外部オーケストレータに割り当てられた名前>
orchestrator_system/workload_type	サービス
orchestrator_system/service_name	<Name of the load balancing virtual server>

生成されるラベル

負荷分散仮想サーバーごとに、外部オーケストレータは次のラベルを生成します。

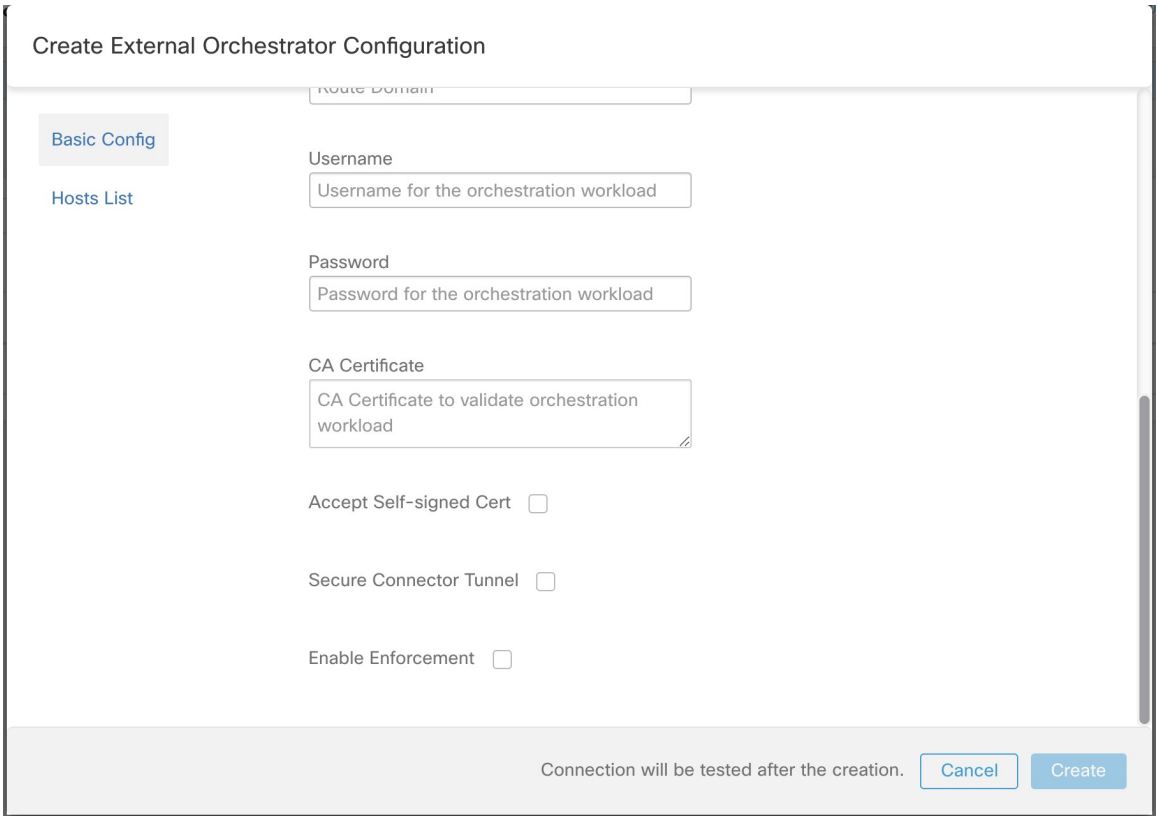
キー	値
orchestrator_annotation/snat_address	<仮想サーバーの SNAT アドレス>

Citrix Netscaler のポリシーの適用

この機能により、Secure Workload は、ラベル付きの *Citrix Netscaler* 仮想サーバーに一致するプロバイダーグループの論理ポリシーを *Citrix Netscaler* ACL ルールに変換し、それらのルールを REST API を使用してロードバランサアプライアンスに展開できます。前述のように、すべての既存の ACL ルールは、Secure Workload によって生成されたポリシールールに置き換えられます。

デフォルトでは、以下の図に示すように、[オーケストレータの作成（Create Orchestrator）] ダイアログで、[適用の有効化（Enable Enforcement）] フィールドはオンになっておらず、適用は無効化されています。

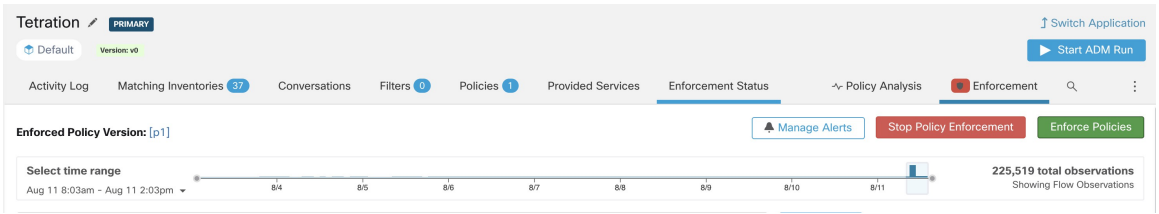
図 *50* : 設定オプション *[*適用の有効化（***Enable Enforcement***）*]*



指定されたチェックボックスをクリックするだけで、オーケストレータの適用を有効にすることができます。このオプションは、必要に応じていつでも変更できます。

オーケストレータ設定の作成または編集によって有効にされたかどうかに関係なく、オーケストレータの適用を有効にしても、現在の論理ポリシーがロード バランサ アプライアンスにすぐに展開されることはありません。このタスクは、次の図に示すように、ユーザーによって、またはインベントリの更新によってトリガーされるワークスペースポリシー適用の一環として実行されます。ただし、オーケストレータの適用を無効にすると、展開されたすべての ACL ルールが *Citrix Netscaler* ロードバランサからすぐに削除されます。

図 *51* : ワークスペースポリシーの適用



（注）

- *Citrix Netscaler* のオーケストレータも、ACL ルールの逸脱を検知し、Secure Workload ポリシーに置き換えます。そのため、ロードバランシング仮想サーバーに対するポリシーの変更は、すべて Secure Workload を使用して行う必要があります。
- ポリシーの適用が停止されるか、外部オーケストレータが削除されると、すべての Secure Workload ポリシーが *Citrix Netscaler* ロードバランサから削除されるため、ACL は空になります。

外部オーケストレータの OpenAPI ポリシー適用ステータスを使用して、外部オーケストレータに関連付けられたロード バランサ アプライアンスへの Secure Workload ポリシー適用のステータスを取得できます。この機能は、*Citrix Netscaler* アプライアンスへの ACL ルールの展開が成功したかどうかを確認するのに役立ちます。

警告

- 適用が有効になっている場合、Secure Workload ポリシーは常に ACL のグローバルリスト（パーティション *default*）に展開されます。
- 単一のアドレスとして指定された VIP のみがサポートされています。言い換えれば、アドレスパターンとして指定された VIP はサポートされていません。
- 検出されたサービス（*Citrix Netscaler* 仮想サーバー）の可視性はサポートされていません。

トラブルシューティング

- 接続の問題 Secure Workload は、Secure Workload アプライアンスサーバーの 1 つから、または TaaS の場合はクラウドから、または Secure Workload Secure Connector トンネル サービスをホストしている VM から、HTTPS 接続を使用して、指定された IP/ホスト名とポート番号に接続しようとします。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。また、指定されたクレデンシャルが正しいこと、および Citrix Netscaler アプライアンスに REST API 要求を送信するための読み取りおよび書き込みアクセス権限があることを確認してください。
- ACL ルールが見つからない ACL ルールが見つからない場合は、ポリシー適用の実行後に、対応する仮想サーバーが有効になっていることを確認してください。そのステータスは稼働状態である必要があります。

TAXII

TAXII（Trusted Automated Exchange of Intelligence Information）統合により、Secure Workload はセキュリティベンダーからの脅威インテリジェンス データ フィードを取り込み、ネットワー

クフローに注釈を付け、悪意のある IP、悪意のあるハッシュなどの STIX（構造化脅威情報表現）インジケータを使用してハッシュを処理できます。

「taxii」タイプの外部オーケストレータ構成が追加されると、Secure Workload アプライアンスは TAXII サーバーへの接続を試み、STIX データフィードコレクションをポーリングします。STIX データフィード（IP とバイナリ ハッシュ インジケータのみ）は解析され、ネットワークフローに注釈を付け、Secure Workload パイプライン内のハッシュを（オーケストレータが構成されているテナントに属するものとして）処理するために使用されます。

インポートされた悪意のある IP と一致するプロバイダーまたはコンシューマのアドレスを持つネットワークフローは、複数値ラベル「orchestrator_malicious_ip_by_<vendor name>」でタグ付けされます。<vendor name> は、ユーザーのオーケストレータ構成入力 TAXII ベンダーであり、ラベル値は「Yes」です。

取り込まれた STIX バイナリ ハッシュ インジケータは、ワークロード プロセス ハッシュに注釈を付けるために使用されます。これは、セキュリティダッシュボード/プロセスハッシュスコアの詳細、およびワークロードプロファイル/ファイルハッシュに表示されます（一致する場合）。

前提条件

- セキュアコネクタトンネル（接続に必要な場合）
- サポートされる TAXII サーバー：1.0
- STIX バージョンでサポートされる TAXII フィード：1.x

設定フィールド

「外部オーケストレータの作成」で説明されている共通の設定フィールドのほかに、次のフィールドを設定できます。

共通フィールド	必須	説明
名前	○	ユーザーが指定したオーケストレーションの名前。
説明	○	オーケストレーションのユーザー指定の説明。
[ベンダー（Vendor）]	○	ベンダーはインテリジェンス データ フィードを提供します。
[フルスナップショット間隔（Full Snapshot Interval）]	○	TAXII フィードの完全なスナップショットを実行する間隔（秒単位）。 （デフォルト：1 日）

共通フィールド	必須	説明
[ポーリングURL（Poll Url）]	○	ポーリングデータへのポーリングの完全な URL パス。
[コレクション（Collection）]	○	ポーリングされる TAXII フィードコレクション名。
[ポーリング日（Poll Days）]	○	TAXII フィードからポーリングする過去の脅威データの数。
[ユーザー名（Username）]		認証用のユーザー名。
[パスワード（Password）]		認証用のパスワード。
[証明書（Certificate）]		認証に使用されるクライアント証明書。
[キー（Key）]		クライアント証明書に対応するキー。
[CA証明書（CA Certificate）]		オーケストレーション エンドポイントを検証する CA 証明書。
[自己署名証明書の受け入れ（Accept Self-signed Cert）]		TAXII API サーバー証明書の厳密な SSL チェックを無効にするチェックボックス にするチェックボックス
[セキュアコネクタトンネル（Secureconnector Tunnel）]		セキュアコネクタトンネルを介したこのオーケストレータのホストへのトンネル接続 へのトンネル接続。
[ホストリスト（Hosts List）]	○	TAXII サーバーを指すホスト名/IP とポートのペア。

ワークフロー

- ユーザーは Secure Workload クラスタから該当 IP/ポートで TAXII サーバーに到達できることを最初に確認する必要があります。
- ポーリングパスと TAXII フィード名を使用して、正しい TAXII サーバーを設定します。

生成されるラベル

キー	値
orchestrator_system/orch_type	*TAXII*
orchestrator_system/cluster_id	Cisco Secure Workload クラスタ設定の UUID。
orchestrator_system/cluster_name	このクラスタの設定に付けられた名前。
orchestrator_malicious_ip_by_<vendor>	フロープロバイダー/コンシューマアドレスがインポートされた TAXII の悪意のある IP データと一致する場合、「Yes」。

警告

- TAXII 統合は、オンプレミスの Cisco Secure Workload のみでサポートされます。
- TAXII フィードからの IP およびハッシュインジケータのみが取り込まれます。
- 取り込まれる IP の最大数は、TAXII フィードあたり 100K（最近更新された数値）です。
- 取り込まれるハッシュの最大数は、すべての TAXII フィードで 500K（最近更新された数値）です。
- STIX バージョン 1.x の TAXII フィードのみがサポートされています。

トラブルシューティング

- 接続の問題

  Secure Workload は、Secure Workload アプライアンスサーバーの 1 つ、または Secure Workload Secure Connector VPN トンネルサービスをホストしている VM から、指定されたポーリング URL パスへの接続を試みます。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。

TAXII オーケストレータのフルポーリングの動作

デフォルトのフルスナップショット間隔は 24 時間です。

フルスナップショット間隔ごとに、Secure Workload は IP とハッシュの TAXII フィードを前述の制限までラベルデータベースにプルします。

Cisco Secure Firewall Management Center

Secure Workload とシスコの Secure Firewall（旧称 Cisco Firepower）の能力を組み合わせることで、以下を利用したセキュリティソリューションを実現できます。

- [セグメンテーション（Segmentation）]

  ファイアウォールベースのセグメンテーションは、ソフトウェアエージェントがインストールされていないワークロードに適しています。ただし、この方法は、エージェントベースのワークロードにも使用できます。ネットワークに入るトラフィック、ネットワークから出るトラフィック、およびネットワーク内のワークロード間のトラフィックに対して、さまざまなポリシーセットを簡単かつ広く適用できます。

- 仮想パッチ適用

  仮想パッチ適用は、ソフトウェアエージェントがインストールされているワークロードに侵入防御システム（IPS）保護を追加します。この統合を使用して、悪意のあるトラフィックがアプリケーションに入ることを防ぎます。仮想パッチ適用設定により、Cisco Secure Workload は、IPS ポリシーの作成中に考慮すべき CVE を FMC に公開します。

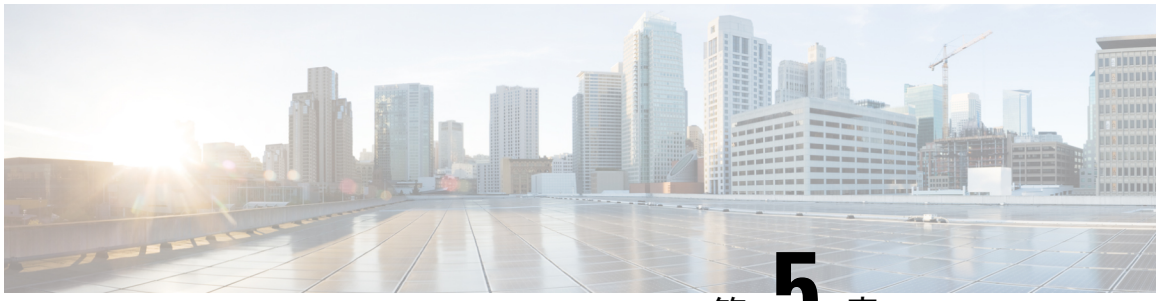
この統合により、Cisco Secure Workload は、Cisco Secure Firewall Management Center インスタンスによって管理される Cisco Secure Firewall Threat Defense（旧称 Firepower Threat Defense）ファイアウォールでセグメンテーションポリシーを自動的に適用および管理します。ポリシーは動的に更新され、アプリケーション環境の変更に応じて、ポリシーが適用する一連のワークロードは継続的に更新されます。

ネットワークインベントリは、セグメンテーションポリシーのベースとなる Cisco Secure Workload インベントリフィルタによって動的に更新されます。ネットワークでワークロードが追加、変更、または削除されると、Cisco Secure Firewall Management Center 内のダイナミックオブジェクトが Cisco Secure Workload により自動的に更新されます。対応するアクセスコントロールルールは、これらのダイナミックオブジェクトに基づきます。適用されたポリシーの変更はすべて、管理対象の Cisco Secure Firewall Threat Defense（旧称 Firepower Threat Defense、FTD）デバイスに自動的に展開されます。Cisco Secure Firewall Management Center で変更を再展開する必要はありません。

動作の詳細、サポートされているプラットフォーム、制限、両方の製品のセットアップ手順、トラブルシューティング情報など、この統合に関する完全な情報については、「Cisco Secure Workload and Cisco Secure Firewall Management Center Integration Guide」を参照してください。

オーケストレータにより生成されるラベル

なし：Cisco Secure Firewall Management Center 外部オーケストレータはユーザー注釈を生成しません。



第 5 章

コネクタ

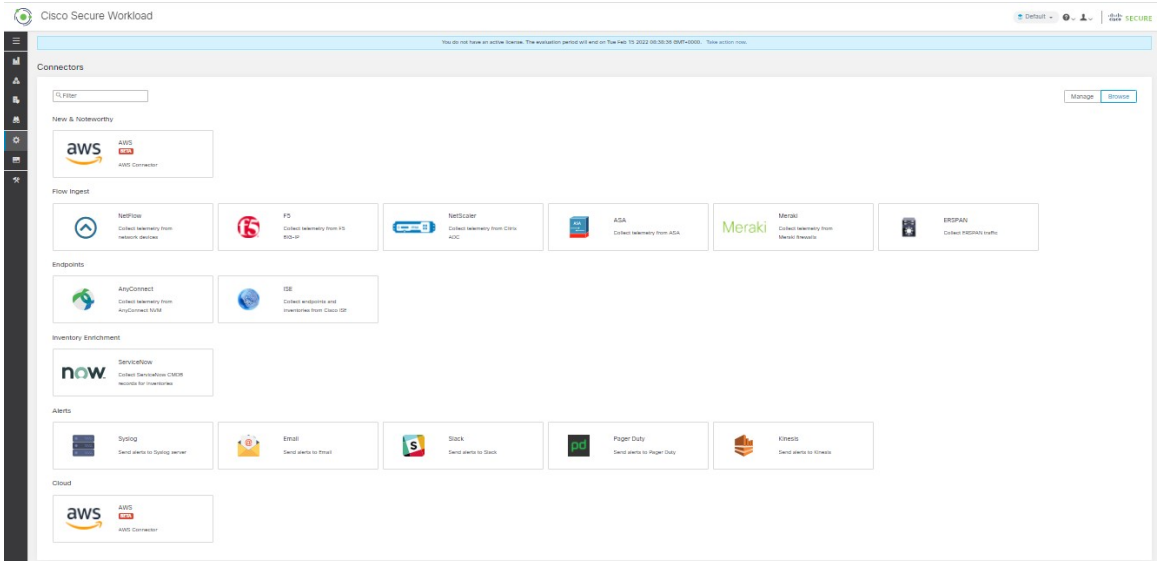
- コネクタとは （217 ページ）
- コネクタ用の仮想アプライアンス （327 ページ）
- コネクタのライフサイクル管理 （340 ページ）
- コネクタおよび仮想アプライアンスの構成管理 （345 ページ）
- トラブルシューティング （361 ページ）
- コネクタアラート （395 ページ）
- Cisco Secure Firewall Management Center （400 ページ）

コネクタとは

コネクタを使用すると、次のようなさまざまな目的で、Secure Workload を他のリソースと統合できます。

- フローの取り込み用のコネクタ
- インベントリ強化用のコネクタ
- Cloud Connector
- エンドポイントのコネクタ
- アラート通知用のコネクタ

図 *52 :* コネクター覧



ほとんどのコネクタには、仮想アプライアンスが必要です。詳細については、「コネクタ用の仮想アプライアンス」を参照してください。

[コネクタ（Connectors）] ページへの移動

コネクタを構成して操作するには、ウィンドウの左側にあるナビゲーションバーで **[管理（Manage）] > [コネクタ（Connectors）]** をクリックします。

フローの取り込み用のコネクタ

さまざまなネットワークスイッチ、ルータ、およびその他のミドルボックス（ロードバランサやファイアウォールなど）から Cisco Secure Workload にフローを取り込むストリームフローを監視するためのコネクタです。Secure Workload は NetFlow v9、IPFIX、およびカスタムプロトコルを介したフローの取り込みをサポートします。ミドルボックスコネクタはフロー観測の他に、クライアント側とサーバー側のフローをつなぎ合わせて、どのクライアントフローがどのサーバーフローに関連しているかを把握します。

コネクタ	説明	仮想アプライアンス上に展開
NetFlow	ルータやスイッチなどのネットワークデバイスから NetFlow V9 や IP-FIX テレメトリを収集します。	Cisco Secure Workload Ingest

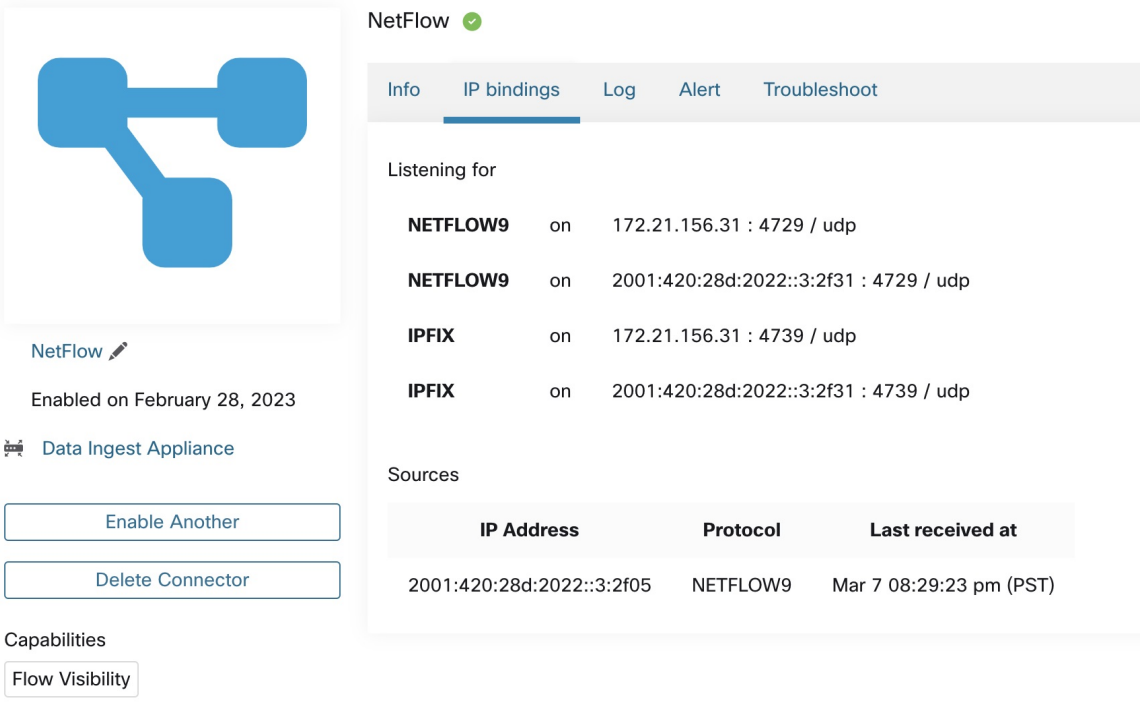
コネクタ	説明	仮想アプライアンス上に展開
F5 BIG-IP	F5 BIG-IP からテレメトリを収集し、クライアント側とサーバー側のフローをつなぎ合わせ、ユーザー属性でクライアントインベントリを充実させます。	Cisco Secure Workload Ingest
Citrix NetScaler	Citrix ADC からテレメトリを収集し、クライアント側とサーバー側のフローをつなぎ合わせます。	Cisco Secure Workload Ingest
Cisco Secure Connector Firewall	Cisco Secure Firewall ASA および Cisco Secure Firewall Threat Defense からテレメトリデータを収集し、クライアント側とサーバー側のフローをつなぎ合わせます。	Cisco Secure Workload Ingest
Meraki	Meraki ファイアウォールからテレメトリデータを収集します。	Cisco Secure Workload Ingest
ERSPAN	ERSPAN をサポートするネットワークデバイスから ERSPAN テレメトリデータを収集します。	Cisco Secure Workload Ingest
関連項目	Cloud Connector	–

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。

NetFlow コネクタ

NetFlow コネクタを使用することにより、Secure Workload は、ネットワーク内のルータおよびスイッチからフロー観測データを取り込むことができます。このソリューションを使用すると、Cisco スイッチにより、処理のために Secure Workload Ingest アプライアンスでホストされている NetFlow コネクタに NetFlow レコードがリレーされるため、ホストでソフトウェアエージェントを実行する必要がありません。

図 *53 : NetFlow* コネクタ



NetFlow とは

NetFlow プロトコルを使用すると、ルータとスイッチは、それらを通過するトラフィックをフローに集約し、これらのフローをフローコレクターにエクスポートできます。フローコレクターはこれらのフローレコードを受け取り、オフラインでのクエリと分析のためにフローストレージに保存します。NetFlow は、ほとんどのシスコ製ルータおよびスイッチでサポートされています。

通常、セットアップには次の手順が含まれます。

1. 1 つ以上のネットワークデバイスで NetFlow 機能を有効にし、デバイスがエクスポートする必要があるフローテンプレートを構成します。
2. リモートネットワークデバイスでNetFlowコレクタのエンドポイント情報を設定します。このNetFlowコレクタが、設定されたエンドポイントでリッスンし、NetFlowフローレコードを受信して処理します。

Cisco Secure Workload へのフローの取り込み

NetFlow コネクタは、本質的に NetFlow コレクタです。コネクタは、ネットワークデバイスからフローレコードを受信し、フロー分析のために Secure Workload に転送します。NetFlow コネクタは、Secure Workload Ingest アプライアンスで有効にし、Docker コンテナとして実行できます。

NetFlow コネクタは、Secure Workload NetFlow エージェントとしても Secure Workload に登録されます。NetFlow コネクタは、NetFlow プロトコルパケット（つまり、フローレコード）のカプセル化を解除し、通常の Secure Workload エージェントのようにフローを処理して報告します。優れた可視性エージェントとは異なり、プロセスやインターフェースの情報は報告しません。

（注） NetFlow コネクタは、NetFlow v9 および IPFIX プロトコルをサポートしています。

（注） 各 NetFlow コネクタは、1 つの VRF のフローのみを報告する必要があります。コネクタによってエクスポートされたフローは、Secure Workload クラスタのエージェント VRF 設定に基づいて VRF に配置されます。コネクタの VRF を設定するには、**[管理（Manage）]** > **[エージェント（Agents）]**に移動し、[設定（Configuration）] タブをクリックします。このページの [エージェントのリモートVRF設定（Agent Remote VRF Configurations）] セクションで、[設定の作成（Create Config）] をクリックし、コネクタに関する詳細を指定します。このフォームで、VRF の名前、コネクタの IP サブネット、およびフローレコードをクラスタに送信できる可能性のあるポート番号の範囲を提供するようにユーザーに要求します。

レート制限

NetFlow コネクタは、毎秒 15000 までのフローを受け入れます。特定の NetFlow v9 または IPFIX パケットには、1 つ以上のフローおよびテンプレートレコードが含まれている可能性があることに注意してください。NetFlow コネクタはパケットを解析し、フローを識別します。コネクタが毎秒 15000 を超えるフローを解析する場合、超過分のフローレコードはドロップされます。

また、フローレートがこの許容限度内にある場合にのみ、Secure Workload カスタマーサポートが NetFlow コネクタをサポートすることにも注意してください。フローレートが毎秒 15000 フローを超える場合は、まずフローを調整して制限内に収めて、少なくとも 3 日間このレベルを維持することをお勧めします（高着信フローレートに関連した問題を回避するため）。元の問題が解決しない場合は、カスタマーサポートが問題の調査を開始し、適切な回避策や解決策を特定します。

サポートされる情報要素

NetFlow コネクタは、NetFlow v9 および IPFIX プロトコルの次の情報要素のみをサポートします。これらの要素の詳細については、「IP フロー情報エクスポート（IPFIX）エンティティ」マニュアルを参照してください。

Element ID	名前	説明	必須
1	octetDeltaCount	このフローの着信パケットのオクテット数。	○

Element ID	名前	説明	必須
2	packetDeltaCount	このフローの着信パケット数。	○
4	protocolIdentifier	IPパケットヘッダーのプロトコル番号の値。	○
6	tcpControlBits	このフローのパケットに対して観測されたTCP制御ビット。エージェントによって処理されるのは、FIN、SYN、RST、PSH、ACK、およびURGフラグのみです。	×
7	sourceTransportPort	トランスポートヘッダー内の送信元ポートID。	○
8	sourceIPv4Address	IPパケットヘッダー内のIPv4送信元アドレス。	8または27のいずれか
11	destinationTransportPort	トランスポートヘッダー内の宛先ポートID。	○
12	destinationIPv4Address	IPパケットヘッダー内のIPv4宛先アドレス。	12または28のいずれか
27	sourceIPv6Address	IPパケットヘッダー内のIPv6送信元アドレス。	8または27のいずれか
28	destinationIPv6Address	IPパケットヘッダーのIPv6宛先アドレス。	12または28
150	flowStartSeconds	フローの先頭パケットの絶対タイムスタンプ（秒単位）。	×
151	flowEndSeconds	フローの最終パケットの絶対タイムスタンプ（秒単位）。	×

Element ID	名前	説明	必須
152	flowStartMilliseconds	フローの先頭パケットの絶対タイムスタンプ（ミリ秒単位）。	×
153	flowEndMilliseconds	フローの最終パケットの絶対タイムスタンプ（ミリ秒単位）。	×
154	flowStartMicroseconds	フローの先頭パケットの絶対タイムスタンプ（マイクロ秒単位）。	×
155	flowEndMicroseconds	フローの最終パケットの絶対タイムスタンプ（マイクロ秒単位）。	×
156	flowStartNanoseconds	フローの先頭パケットの絶対タイムスタンプ（ナノ秒単位）。	×
157	flowEndNanoseconds	フローの最終パケットの絶対タイムスタンプ（ナノ秒単位）。	×

スイッチでの NetFlow の設定方法

次の手順は、Nexus 9000 スイッチ用です。他のシスコプラットフォームでは、設定方法が若干異なる場合があります。いずれの場合も、設定するシスコプラットフォームに関する公式のシスコ コンフィグレーション ガイドも参照してください。

手順

ステップ 1 グローバル コンフィギュレーション モードを開始します。

```
switch# configure terminal
```

ステップ 2 NetFlow 機能を有効にします。

```
switch(config)# feature netflow
```

ステップ 3 フローレコードを設定します。

次の設定例は、NetFlow レコードでフローの 5 つのタプル情報を生成する方法を示しています。

```
switch(config)# flow record ipv4-records
switch(config-flow-record)# description IPv4Flow
switch(config-flow-record)# match ipv4 source address
switch(config-flow-record)# match ipv4 destination address
```

```
switch(config-flow-record)# match ip protocol
switch(config-flow-record)# match transport source-port
switch(config-flow-record)# match transport destination-port
switch(config-flow-record)# collect transport tcp flags
switch(config-flow-record)# collect counter bytes
switch(config-flow-record)# collect counter packets
```

ステップ 4 フローエクスポータを設定します。

次の設定例では、NetFlow プロトコルバージョン、NetFlow テンプレート交換間隔、および NetFlow コレクタエンドポイントの詳細を指定しています。Secure Workload Ingest アプライアンスで NetFlow コネクタが有効になっている IP とポートを指定してください。

```
switch(config)# flow exporter flow-exporter-one
switch(config-flow-exporter)# description NetFlowv9ToNetFlowConnector
switch(config-flow-exporter)# destination 172.26.230.173 use-vrf management
switch(config-flow-exporter)# transport udp 4729
switch(config-flow-exporter)# source mgmt0
switch(config-flow-exporter)# version 9
switch(config-flow-exporter-version-9)# template data timeout 20
```

ステップ 5 フローモニターを設定します。

フローモニターを作成して、フローレコードおよびフローエクスポータと関連付けます。

```
switch(config)# flow monitor ipv4-monitor
switch(config-flow-monitor)# description IPv4FlowMonitor
switch(config-flow-monitor)# record ipv4-records
switch(config-flow-monitor)# exporter flow-exporter-one
```

ステップ 6 フローモニターをインターフェイスに適用します。

```
switch(config)# interface Ethernet 1/1
switch(config-if)# ip flow monitor ipv4-monitor input
```

上記の手順により、インターフェイス 1/1 を通過する入力トラフィックの NetFlow v9 プロトコルパケットをエクスポートするように、Nexus 9000 の NetFlow が設定されます。フローレコードは、UDP プロトコルを使用して 172.26.230.173:4729 に送信されます。各フローレコードには、トラフィックの 5 つのタプル情報と、フローのバイト数/パケット数が含まれます。

次のスクリーンショットは、Nexus 9000 スイッチにおける NetFlow の実行コンフィギュレーションを示しています。

図 *54 : Cisco Nexus 9000* スイッチでの *NetFlow* の実行コンフィギュレーション

```
switch# show running-config netflow

!Command: show running-config netflow
!Time: Wed Mar 21 04:25:21 2018

version 7.0(3)I7(1)
feature netflow

flow timeout 60
flow exporter flow-exporter-173
  destination 172.26.230.173 use-vrf management
  transport udp 4729
  source mgmt0
  version 9
    template data timeout 20
flow record ipv4-records
  match ipv4 source address
  match ipv4 destination address
  match ip protocol
  match transport source-port
  match transport destination-port
  collect transport tcp flags
  collect counter bytes
  collect counter packets
  collect timestamp sys-uptime first
  collect timestamp sys-uptime last
flow monitor ipv4-monitor
  record ipv4-records
  exporter flow-exporter-173


interface Ethernet1/1
  ip flow monitor ipv4-monitor input

interface Ethernet1/2
  ip flow monitor ipv4-monitor input

switch#
```

コネクタの設定方法

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。NetFlow コネクタの場合、IPv4 および IPv6（デュアルスタックモード）アドレスがサポートされます。ただし、デュアルスタックのサポートはベータ機能であることに注意してください。

コネクタでは、次の項目を設定できます。

- ログ：詳細については、「ログ設定」を参照してください。

さらに、許可されたコマンドを使用して、Secure Workload Ingest アプライアンスの Docker コンテナで、コネクタの IPFIX プロトコルのリスニングポートを更新できます。このコマンドは、コネクタのコネクタ ID、更新するポートのタイプ、および新しいポートの情報が提供されると、アプライアンスで発行できます。コネクタ ID は、Secure Workload UI の [コネクタ（connector）] ページにあります。詳細については、update-listening-ports を参照してください。

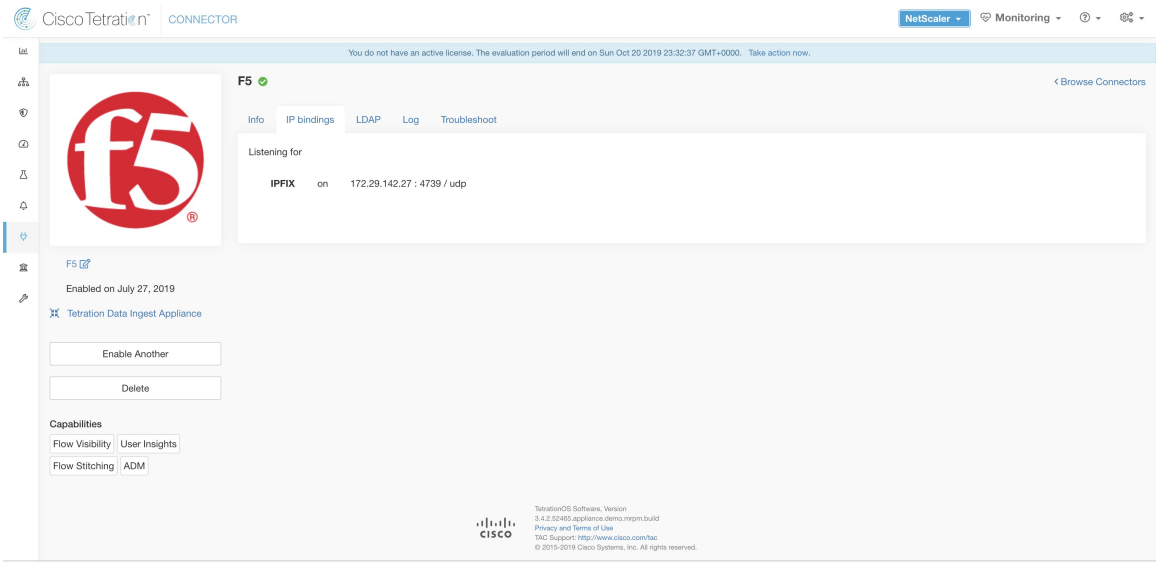
制限

メトリック	制限
1 つの Secure Workload Ingest アプライアンス上の NetFlow コネクタの最大数	3
1 つのテナント（ルート範囲）上の NetFlow コネクタの最大数	10
Secure Workload 上の NetFlow コネクタの最大数	100

F5 コネクタ

F5 コネクタにより、Secure Workload は F5 BIG-IP ADC からのフロー観測を取り込むことができます。Secure Workload は F5 BIG-IP ADC のフロー観測をリモートで監視し、クライアント側とサーバー側のフローをスティッチングし、クライアント IP でユーザーに注釈をつけることが可能です（ユーザー情報が利用可能な場合）。このソリューションを使用すると、F5 BIG-IP ADC は処理目的で IPFIX レコードを F5 コネクタにエクスポートするように設定されるため、ホストはソフトウェアエージェントを実行する必要がありません。

図 *55 : F5* コネクタ



F5 BIG-IP IPFIX について

F5 BIG-IP IPFIX ロギングは、F5 BIG-IP を通過するトラフィックのフローデータを収集し、IPFIX レコードをフローコレクタにエクスポートします。

通常、セットアップには次の手順が含まれます。

1. F5 BIG-IP アプライアンスで IPFIX Log-Publisher を作成します。
2. F5 BIG-IP アプライアンスで IPFIX Log-Destination を構成します。この log-destination は、設定されたエンドポイントでリッスンし、フローレコードを受信して処理します。
3. IPFIX フローレコードを log-publisher に公開する F5 iRule を作成します。
4. F5 iRule を対象の仮想サーバーに追加します。

（注） F5 コネクタは、F5 BIG-IP ソフトウェアバージョン 12.1.2 以降をサポートします。

Cisco Secure Workload へのフローの取り込み

F5 BIG-IP コネクタは、本質的に IPFIX コレクタです。このコネクタは、F5 BIG-IP ADC からフローレコードを受信すると、NATed フローをステッチして、フロー分析の目的で Secure Workload に転送します。さらに、F5 コネクタに LDAP が設定されている場合、トランザクションに関連付けられたユーザーの設定済み LDAP 属性の値が判断されます（F5 がトランザクションを処理する前にユーザーを認証する場合）。LDAP 属性は、フローが発生したクライアントの IP アドレスに関連付けられています。

（注） F5 コネクタは IPFIX プロトコルのみをサポートします。

（注） 各 F5 コネクタは、1 つの VRF のフローのみを報告する必要があります。コネクタによってエクスポートされたフローは、Cisco Secure Workload クラスタのエージェント VRF 設定に基づいて VRF に配置されます。コネクタの VRF を設定するには、**[管理（Manage）]** > **[エージェント（Agents）]**に移動し、[設定（Configuration）] タブをクリックします。このページの [エージェントのリモートVRF設定（Agent Remote VRF Configurations）] セクションで、[設定の作成（Create Config）] をクリックし、コネクタに関する詳細を指定します。このフォームで、VRF の名前、コネクタの IP サブネット、およびフローレコードをクラスタに送信できる可能性のあるポート番号の範囲を提供するようにユーザーに要求します。

F5 BIG-IP での IPFIX の設定方法

次の手順は、F5 BIG-IP ロードバランサ用です。（参照：IPFIX 用 F5 BIG-IP を設定する）

目的	説明
1. IPFIX コレクタのプールを作成します。	F5 BIG-IP アプライアンスで、IPFIX コレクタのプールを作成します。これらは、Secure Workload Ingest アプライアンスの F5 コネクタに関連付けられた IP アドレスです。F5 コネクタは、VM 上の Docker コンテナで実行され、ポート 4739 で IPFIX パケットをリッスンします。
2. log-destination を作成します。	F5 BIG-IP アプライアンスのログ宛先設定は、使用する必要がある IPFIX コレクタの実際のプールを指定します。
3. log-publisher を作成します。	ログパブリッシャは、F5 BIG-IP が IPFIX メッセージを送信する場所を指定します。パブリッシャは log-destination にバインドされています。
4. F5 と Secure Workload の承認済み iRule を追加します。	Cisco Secure Workload と F5 は、フローレコードを F5 コネクタにエクスポートする iRules を開発しました。これらの iRule は、すべてのエンドポイント、バイト数とパケット数、フローの開始時間と終了時間（ミリ秒単位）など、特定のトランザクションに関する完全な情報をエクスポートします。F5 コネクタは 4 つの独立したフローを作成し、各フローを関連するフローと照合します。

目的	説明
5. iRule を仮想サーバーに追加します。	仮想サーバーの iRule 設定で、Cisco Secure Workload 承認済み iRule を仮想サーバーに追加します。

上記の手順では、F5 BIG-IP ロードバランサで IPFIX を設定し、アプライアンスを通過するトラフィックの IPFIX プロトコルパケットをエクスポートします。F5 の設定例を示します。

図 *56 : F5 BIG-IP* ロードバランサでの *IPFIX* の設定の実行

```
root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmos)# show running-config ltm virtual vip-1 rules
ltm virtual vip-1 {
    rules {
        ipfix-rule-1
    }
}
root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmos)# show running-config ltm  pool ipfix-pool-1
ltm pool ipfix-pool-1 {
    members {
        10.28.118.6:ipfix {
            address 10.28.118.6
            session monitor-enabled
            state up
        }
    }
    monitor gateway_icmp
}
root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmos)# show running-config ltm virtual vip-1 rules
ltm virtual vip-1 {
    rules {
        ipfix-rule-1
    }
}
root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmos)# show running-config sys  log-config
sys log-config destination ipfix ipfix-collector-1 {
    pool-name ipfix-pool-1
    transport-profile udp
}
sys log-config publisher ipfix-pub-1 {
    destinations {
        ipfix-collector-1 { }
    }
}
root@(localhost)(cfg-sync Standalone)(Active)(/Common)(tmos)#
```

上記の例では、フローレコードは *ipfix-pub-1* に公開されます。*ipfix-pub-1* には、IPFIX メッセージを IPFIX プール *ipfix-pool-1* に送信する log-destination *ipfix-collector-1* が設定されています。*ipfix-pool-1* には、IPFIX コレクタの 1 つとして 10.28.118.6 が設定されています。仮想サーバー *vip-1* は、IPFIX テンプレートとテンプレートの入力方法と送信方法を指定する IPFIX iRule *ipfix-rule-1* を使用して設定されます。

- TCP 仮想サーバーの F5 および Secure Workload 承認済み iRule は、次のファイルにあります。「TCP 仮想サーバーの L4 iRule」を参照してください。

UDP 仮想サーバーの F5 および Secure Workload 承認済み iRule は、次のファイルにあります。

- 「UDP 仮想サーバーの L4 iRule 」を参照してください。

認証が有効になっている HTTPS 仮想サーバーの F5 および Secure Workload 承認済み iRule は、次のファイルにあります。

- 「HTTPS 仮想サーバーの iRule」を参照してください。

（注） このガイドからダウンロードした iRule を使用する前に、iRule を追加する F5 コネクタで設定された log-publisher を指すように **log-publisher** を更新してください。

（注） F5 は GitHub リポジトリ f5-tetration を公開して、ユーザーがフロースティッチングを開始できるようにしています。さまざまなプロトコルタイプの F5 コネクタに IPFIX レコードを公開する iRules は、f5-tetration/irules で入手できます。最新の iRule 定義については、このサイトにアクセスしてください。さらに、F5 は次のためのスクリプトも開発しました。（1）仮想サーバーに正しい iRule をインストールする、（2）IPFIX コレクタエンドポイント（F5 コネクタが IPFIX レコードをリッスンする）のプールを追加する、（3）log-collector と log-publisher を設定する、（4）正しい iRule を仮想サーバーにバインドする。このツールにより、フロースティッチングのユースケースを有効にしながら、手作業の設定とユーザーエラーを最小限に抑えます。スクリプトは f5-tetration/scripts で入手できます。

コネクタの設定方法

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。

コネクタでは、以下を設定できます。

- LDAP：LDAP 設定は、LDAP 属性の検出をサポートし、ユーザー名に対応する属性を選択するワークフローと、ユーザーごとに取得される最大 6 つの属性のリストを提供します。詳細については、「検出」を参照してください。
- ログ：詳細については、「ログ設定」を参照してください。

さらに、コンテナの実行が許可されたコマンドを使用して、Secure Workload Ingest アプライアンスの Docker コンテナで、コネクタの IPFIX プロトコルのリスニングポートを更新できます。このコマンドは、コネクタのコネクタ ID、更新するポートのタイプ、および新しいポートの情報が提供されると、アプライアンスで発行できます。コネクタ ID は、Secure Workload UI の [コネクタ（connector）] ページにあります。詳細については、update-listening-ports を参照してください。

制限

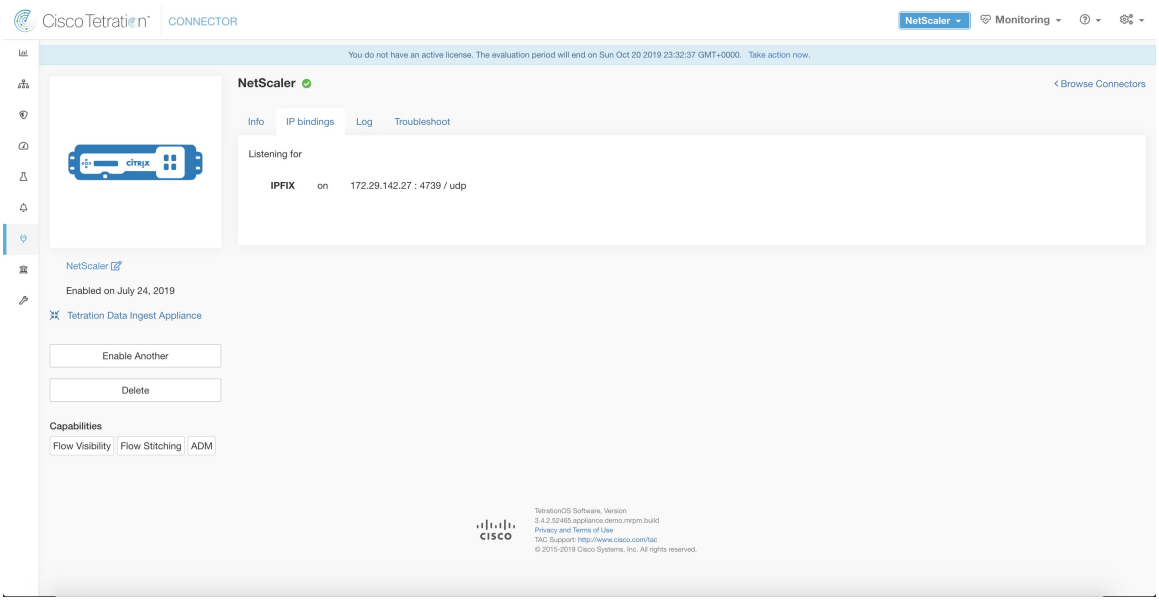
メトリック	制限
1 つの Secure Workload Ingest アプライアンスにおける F5 コネクタの最大数	3

メトリック	制限
1 つのテナント（ルート範囲）における F5 コネクタの最大数	10
Cisco Secure Workload における F5 コネクタの最大数	100

NetScaler コネクタ

Secure Workload は NetScaler コネクタを使用して Citrix ADC（Citrix NetScaler）からフロー観測データを取り込むことができます。これにより、Secure Workload は Citrix ADC のフロー観測をリモートで監視し、クライアント側とサーバー側のフローをつなぎ合わせることができます。このソリューションを使用すると、Citrix ADC は処理目的で IPFIX レコードを NetScaler コネクタにエクスポートするように設定されるため、ホストはソフトウェアエージェントを実行する必要がありません。

図 *57 : NetScaler* コネクタ



Citrix NetScaler AppFlow とは

Citrix NetScaler AppFlow は、NetScaler を通過するトラフィックのフローデータを収集し、IPFIX レコードをフローコレクタにエクスポートします。Citrix AppFlow プロトコルは、IPFIX を使用してフローをフローコレクタにエクスポートします。Citrix AppFlow は、Citrix NetScaler ロードバランサでサポートされています。

通常、セットアップには次の手順が含まれます。

1. 1 つ以上の Citrix NetScaler インスタンスで AppFlow 機能を有効にします。

2. リモートネットワークデバイスでAppFlowコレクタのエンドポイント情報を設定します。このAppFlowコレクタが、設定されたエンドポイントでリッスンし、フローレコードを受信して処理します。
3. AppFlowのアクションとポリシーを設定して、フローレコードをAppFlowコレクタにエクスポートします。

（注） NetScaler コネクタは、Citrix ADC ソフトウェアバージョン 11.1.51.26 以降をサポートしています。

Cisco Secure Workload へのフローの取り込み

NetScalerコネクタは、本質的にCitrix AppFlow（IPFIX）コレクタです。コネクタはCitrix ADCからフローレコードを受信すると、NATedフローをステッチして、フロー分析の目的でSecure Workloadに転送します。NetScaler コネクタは Cisco Ingest Secure Workload アプライアンスで有効にすることができ、Docker コンテナとして稼働します。また、NetScaler コネクタは Secure Workload NetScaler エージェントとして Secure Workload に登録されます。

（注） NetScaler コネクタは、IPFIX プロトコルのみをサポートします。

（注） 各NetScalerコネクタは、1つのVRFのフローのみを報告する必要があります。コネクタによってエクスポートされたフローは、Secure Workload クラスタ内のエージェント VRF 設定に基づいて VRF に配置されます。コネクタの VRF を設定するには、**[管理（Manage）]** > **[エージェント（Agents）]**に移動し、[設定（Configuration）] タブをクリックします。このページの [エージェントのリモートVRF設定（Agent Remote VRF Configurations）] セクションで、[設定の作成（Create Config）] をクリックし、コネクタに関する詳細を指定します。このフォームで、VRFの名前、コネクタの IP サブネット、およびフローレコードをクラスタに送信できる可能性のあるポート番号の範囲を提供するようにユーザーに要求します。

NetScaler での AppFlow の設定方法

次の手順は、NetScaler ロードバランサ用です （「AppFlow の設定」を参照）。

手順

ステップ 1 NetScaler で AppFlow を有効にします。

```
enable ns feature appflow
```

ステップ 2 AppFlow コレクタエンドポイントを追加します。

コレクターは、NetScaler から AppFlow レコードを受け取ります。Secure Workload Ingest アプライアンスで有効になっている NetScaler コネクタの IP とポートを AppFlow コレクタとして指定してください。

```
add appflow collector c1 -IPAddress 172.26.230.173 -port 4739
```

ステップ 3 AppFlow アクションを設定します。

これは、関連付けられた AppFlow ポリシーが一致した場合に AppFlow レコードを取得するコレクタのリストです。

```
add appflow action a1 -collectors c1
```

ステップ 4 AppFlow ポリシーを設定します。

これは、AppFlow レコードを生成するために一致する必要があるルールです。

```
add appflow policy p1 CLIENT.TCP.DSTPORT(22) a1
add appflow policy p2 HTTP.REQ.URL.SUFFIX.EQ("jpeg") a1
```

ステップ 5 AppFlow ポリシーを仮想サーバーにバインドします。

仮想サーバー（VIP）の IP に到達するトラフィックは、AppFlow ポリシーと一致しているか評価されます。一致すると、フローレコードが生成され、関連する AppFlow アクションにリストされているすべてのコレクタに送信されます。

```
bind lb vserver lb1 -policyname p1 -priority 10
```

ステップ 6 必要に応じて、AppFlow ポリシーをグローバルにバインドします（すべての仮想サーバーに対して）。

AppFlow ポリシーは、すべての仮想サーバーにグローバルにバインドすることもできます。このポリシーは、Citrix ADC を通過するすべてのトラフィックに適用されます。

```
bind appflow global p2 1 NEXT -type REQ_DEFAULT
```

ステップ 7 必要に応じてテンプレート更新間隔を設定します。

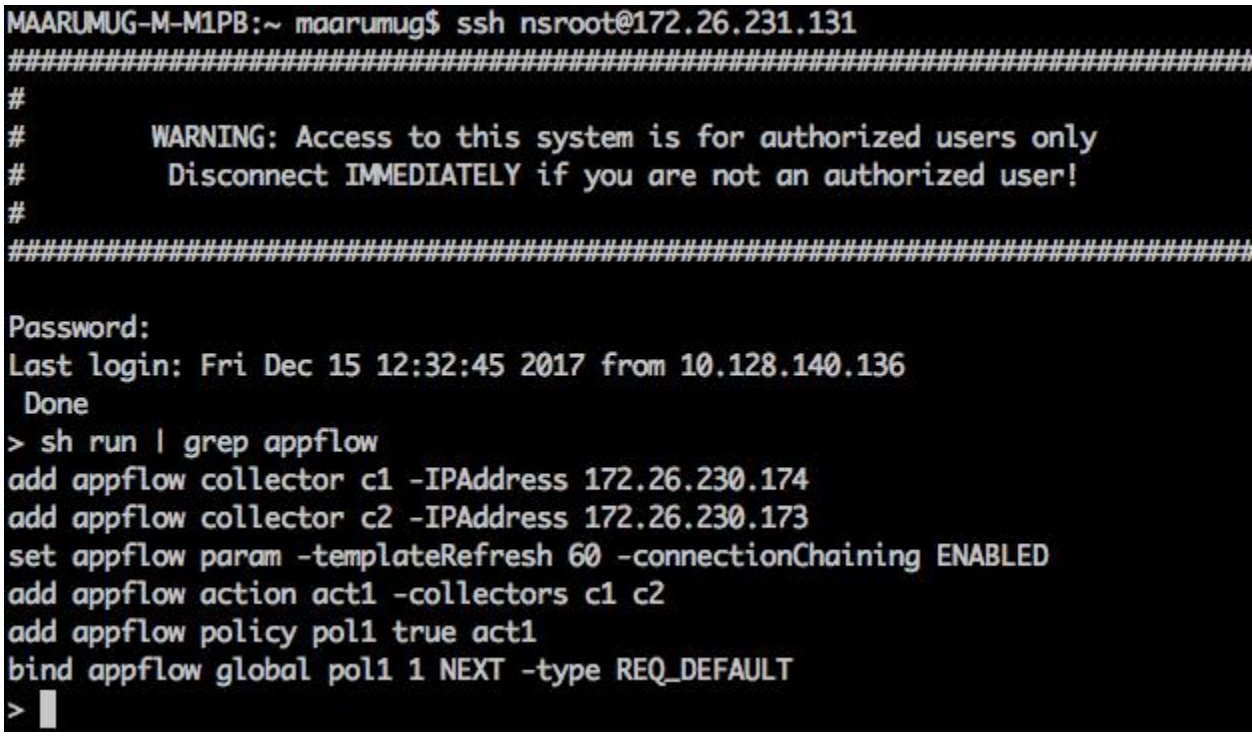
テンプレート更新のデフォルト値は 60 秒です。

```
set appflow param -templatereferesh 60
```

上記の手順により、Citrix NetScaler ロードバランサで AppFlow が設定され、NetScaler を通過するトラフィックの IPFIX プロトコルパケットがエクスポートされます。フローレコードは、172.26.230.173:4739（Vserver lb1 を通過するトラフィックの場合）と 172.26.230.184:4739（NetScaler を通過するすべてのトラフィックの場合）のいずれかに送信されます。各フローレコードには、トラフィックの 5 つのタプル情報と、フローのバイト数/パケット数が含まれます。

次のスクリーンショットは、Citrix NetScaler ロードバランサにおける AppFlow の実行コンフィギュレーションを示しています。

図 *58 : Citrix NetScaler* ロードバランサにおける *AppFlow* の実行コンフィギュレーション



コネクタの設定方法

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。コネクタでは、以下を設定できます。

- ログ：詳細については、「ログ設定」を参照してください。

さらに、許可されたコマンドを使用して、Secure Workload Ingest アプライアンスの Docker コンテナで、コネクタの IPFIX プロトコルのリスニングポートを更新できます。このコマンドは、コネクタのコネクタ ID、更新するポートのタイプ、および新しいポートの情報が提供されると、アプライアンスで発行できます。コネクタ ID は、Secure Workload UI の [コネクタ（connector）] ページにあります。詳細については、update-listening-ports を参照してください。

制限

表 *19 :* 制限

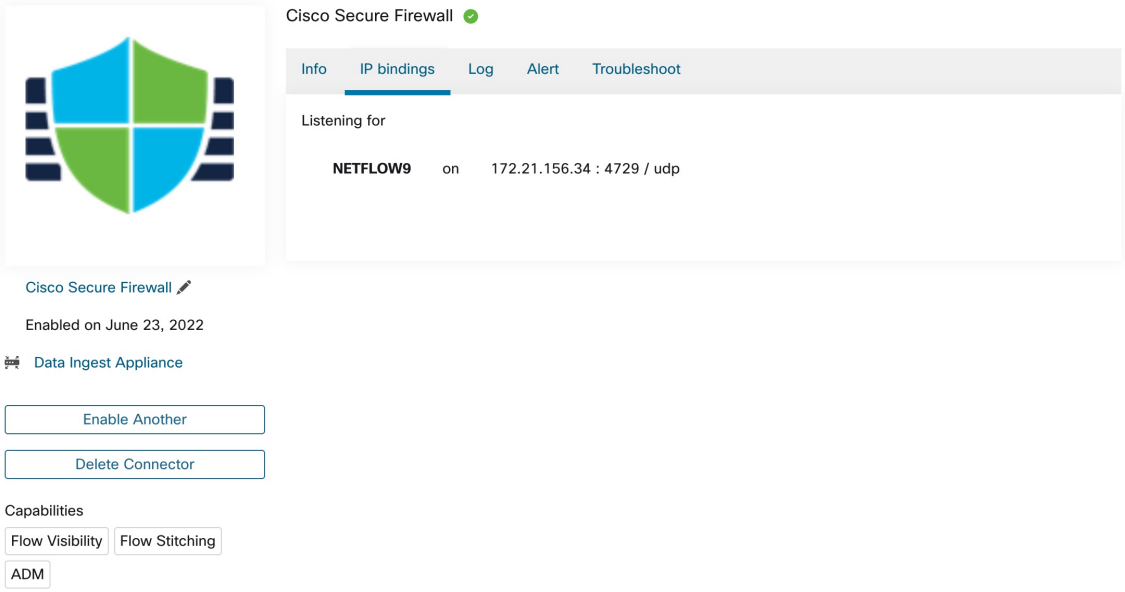
メトリック	制限
1 つの Secure Workload Ingest アプライアンス上の NetScaler コネクタの最大数	3

メトリック	制限
1 つのテナント（ルート範囲）における NetScaler コネクタの最大数	10
Cisco Secure Workload 上の NetScaler コネクタの最大数	100

Cisco Secure Firewall コネクタ

Cisco Secure Firewall コネクタ (旧称 ASA コネクタ）により、Secure Workload は Cisco Secure Firewall ASA（旧称 Cisco ASA）および Cisco Secure Firewall Threat Defense（旧称 Firepower Threat Defense または FTD）からのフロー観測データを取り込むことができます。このソリューションを使用すると、ホストはソフトウェアエージェントを実行する必要がありません。Cisco スイッチが NetFlow セキュアイベントロギング（NSEL）レコードを処理するために Secure Workload Ingest アプライアンスでホストされている Cisco Secure Firewall コネクタにリレーします。

図 *59 : Cisco Secure Firewall* コネクタ



Cisco Secure Firewall ASA NetFlow セキュアイベントロギング（NSEL） は、フロー内の重要なイベントを NetFlow コレクタにエクスポートするステートフルな IP フローモニタリングを提供します。イベントによってフローの状態が変化すると、NSEL イベントがトリガーされ、フロー観測データが状態の変化を引き起こしたイベント情報とともに NetFlow コレクタに送信されます。フローコレクタはこれらのフローレコードを受け取ると、オフラインでクエリと分析を行えるようにフローストレージに保存します。

通常、セットアップには次の手順が含まれます。

1. Cisco Secure Firewall ASA と Cisco Secure Firewall Threat Defense のどちらか、または両方で NSEL 機能を有効にします。
2. Cisco Secure Firewall ASA と Cisco Secure Firewall Threat Defense のどちらかまたは両方で、Cisco Secure Firewall コネクタのエンドポイント情報を設定します。Cisco Secure Firewall コネクタは、設定されたエンドポイントでリッスンして、NSEL レコードを受信および処理します。

Cisco Secure Workload へのフローの取り込み

Cisco Secure Firewall コネクタは、基本的に NetFlow コレクタです。コネクタは、Cisco Secure Firewall ASA および Cisco Secure Firewall Threat Defense から NSEL レコードを受信し、フロー分析のために Secure Workload に転送します。Cisco Secure Firewall コネクタは、Secure Workload Ingest アプライアンスで有効にし、Docker コンテナとして実行できます。

Cisco Secure Firewall コネクタも Secure Workload エージェントとして Secure Workload に登録されます。Cisco Secure Firewall コネクタは、NSEL プロトコルパケット（つまり、フローレコード）のカプセル化を解除し、通常の Secure Workload エージェントのようにフローを処理して報告します。優れた可視性エージェントとは異なり、プロセスやインターフェースの情報は報告しません。

（注） Cisco Secure Firewall コネクタは、NetFlow v9 プロトコルをサポートします。

（注） 各 Cisco Secure Firewall コネクタは、1 つの VRF のフローのみを報告する必要があります。コネクタによってエクスポートされたフローは、Secure Workload クラスタのエージェント VRF 設定に基づいて VRF に配置されます。コネクタの VRF を設定するには、**[管理（Manage）]** > **[エージェント（Agents）]**に移動し、[設定（Configuration）] タブをクリックします。このページの [エージェントのリモートVRF設定（Agent Remote VRF Configurations）] セクションで、[設定の作成（Create Config）] をクリックし、コネクタに関する詳細を指定します。このフォームで、VRF の名前、コネクタの IP サブネット、およびフローレコードをクラスタに送信できる可能性のあるポート番号の範囲を提供するようにユーザーに要求します。

NSEL イベントの処理

次の表は、さまざまな NSEL イベントが Cisco Secure Firewall コネクタによってどのように処理されるのかを示しています。これらの要素の詳細については、『IP Flow Information Export (IPFIX) Entities』ドキュメント [英語] を参照してください。

フローイベント要素 ID：233 要素名：*NF_F_FW_EVENT*	拡張フローイベント要素 ID：33002 要素名：*NF_F_FW_EXT_EVENT*	Cisco Secure Firewall コネクタでのアクション
0（デフォルト、この値を無視）	考慮しない	なし

フローイベント要素 ID：233 要素名：*NF_F_FW_EVENT*	拡張フローイベント要素 ID：33002 要素名：*NF_F_FW_EXT_EVENT*	Cisco Secure Firewall コネクタでのアクション
1（フロー作成）	考慮しない	Cisco Secure Workload にフローを送信
2（フロー削除）	2000 超 (終了理由を示す)	Cisco Secure Workload にフローを送信
3（フロー拒否）	1001（入力 ACL による拒否）	処理拒否とマークされたフローを Cisco Secure Workload に送信
	1002（入力 ACL による拒否）	
	1003（ASA インターフェイスによる接続拒否、またはデバイスへの ICMP(v6) サービス拒否）	
	1004（TCP の最初のパケットが SYN ではない）	
4（フローアラート）	考慮しない	なし
5（フロー更新）	考慮しない	Cisco Secure Workload にフローを送信

Cisco Secure Firewall コネクタは、NSEL レコードに基づいてフロー観測データを Cisco Secure Workload に送信します。NSEL フローレコードは双方向レコードです。したがって、Cisco Secure Firewall コネクタは、Cisco Secure Workload に対してフォワードフローとリバースフローの 2 つのフローを送信します。

以下は、Cisco Secure Firewall コネクタから Cisco Secure Workload に送信されるフロー観測データの詳細です。

フォワードフロー観測データ

フィールド	NSEL 要素 ID	NSEL 要素名
[プロトコル（Protocol）]	4	*NF_F_PROTOCOL*
[送信元アドレス（Source Address）]	8	*NF_F_SRC_ADDR_IPV4*
	27	*NF_F_SRC_ADDR_IPV6*
[送信元ポート（Source Port）]	7	*NF_F_SRC_PORT*
[宛先アドレス（Destination Address）]	12	*NF_F_DST_ADDR_IPV4*
	36	*NF_F_DST_ADDR_IPV6*

フィールド	NSEL 要素 ID	NSEL 要素名
[宛先ポート（Destination Port）]	11	*NF_F_DST_PORT*
[フロー開始時刻（Flow Start Time）]	152	*NF_F_FLOW_CREATE_TIME_MSEC*
[バイト数（Byte Count）]	231	*NF_F_FWD_FLOW_DELTA_BYTES*
[パケット数（Packet Count）]	298	*NF_F_FWD_FLOW_DELTA_PACKETS*

リバースフロー観測情報

フィールド	NSEL 要素 ID	NSEL 要素名
[プロトコル（Protocol）]	4	*NF_F_PROTOCOL*
[送信元アドレス（Source Address）]	12	*NF_F_DST_ADDR_IPV4*
	36	*NF_F_DST_ADDR_IPV6*
[送信元ポート（Source Port）]	11	*NF_F_DST_PORT*
[宛先アドレス（Destination Address）]	8	*NF_F_SRC_ADDR_IPV4*
	27	*NF_F_SRC_ADDR_IPV6*
[宛先ポート（Destination Port）]	7	*NF_F_SRC_PORT*
[フロー開始時刻（Flow Start Time）]	152	*NF_F_FLOW_CREATE_TIME_MSEC*
[バイト数（Byte Count）]	232	*NF_F_REV_FLOW_DELTA_BYTES*
[パケット数（Packet Count）]	299	*NF_F_REV_FLOW_DELTA_PACKETS*

NAT

クライアントから ASA へのフローが NATed の場合、NSEL フローレコードは、サーバー側の NATed IP やポートを示します。Cisco Secure Firewall コネクタは、この情報を使用して、サーバーから ASA へのフロー、および ASA からクライアントへのフローをスティッチングします。

順方向の NATed フローレコードは次のとおりです。

フィールド	NSEL 要素 ID	NSEL 要素名
[プロトコル（Protocol）]	4	*NF_F_PROTOCOL*

フィールド	NSEL 要素 ID	NSEL 要素名
[送信元アドレス（Source Address）]	225	*NF_F_XLATE_SRC_ADDR_IPV4*
	281	*NF_F_XLATE_SRC_ADDR_IPV6*
[送信元ポート（Source Port）]	227	*NF_F_XLATE_SRC_PORT*
[宛先アドレス（Destination Address）]	226	*NF_F_XLATE_DST_ADDR_IPV4*
	282	*NF_F_XLATE_DST_ADDR_IPV6*
[宛先ポート（Destination Port）]	228	*NF_F_XLATE_DST_PORT*
[フロー開始時刻（Flow Start Time）]	152	*NF_F_FLOW_CREATE_TIME_MSEC*
[バイト数（Byte Count）]	231	*NF_F_FWD_FLOW_DELTA_BYTES*
[パケット数（Packet Count）]	298	*NF_F_FWD_FLOW_DELTA_PACKETS*

フォワードフローは、順方向の NATed フローレコード関連としてマーク付けされます（逆も同様）。

逆方向の NATed フローレコードは次のとおりです。

フィールド	NSEL 要素 ID	NSEL 要素名
[プロトコル（Protocol）]	4	*NF_F_PROTOCOL*
[送信元アドレス（Source Address）]	226	*NF_F_XLATE_DST_ADDR_IPV4*
	282	*NF_F_XLATE_DST_ADDR_IPV6*
[送信元ポート（Source Port）]	228	*NF_F_XLATE_DST_PORT*
[宛先アドレス（Destination Address）]	225	*NF_F_XLATE_SRC_ADDR_IPV4*
	281	*NF_F_XLATE_SRC_ADDR_IPV6*
[宛先ポート（Destination Port）]	227	*NF_F_XLATE_SRC_PORT*
[フロー開始時刻（Flow Start Time）]	152	*NF_F_FLOW_CREATE_TIME_MSEC*
[バイト数（Byte Count）]	232	*NF_F_REV_FLOW_DELTA_BYTES*
[パケット数（Packet Count）]	299	*NF_F_REV_FLOW_DELTA_PACKETS*

リバースフローは、逆方向の NATed フローレコード関連としてマーク付けされます（逆も同様）。

（注） このセクションに記載されている NSEL 要素 ID のみが、Cisco Secure Firewall コネクタでサポートされます。

TCP フラグのヒューリスティック

NSEL レコードには、TCP フラグ情報がありません。Cisco Secure Firewall コネクタは、次のヒューリスティックを使用して TCP フラグを設定し、自動ポリシー検出によってフローをさらに分析できるようにします。

- フォワードパケットが 1 つ以上ある場合、SYN をフォワードフローの TCP フラグに追加します。
- フォワードパケットが 2 つ以上あり、リバースパケットが 1 つある場合、フォワードフローの TCP フラグに ACK を追加し、リバースフローの TCP フラグに SYN-ACK を追加します。
- 前の条件が当てはまり、フローイベントがフロー削除の場合、フォワードフローとリバースフローの両方の TCP フラグに FIN を追加します。

Cisco Secure Firewall ASA での NSEL の設定方法

次の手順は、NSEL を設定し、NetFlow パケットをコレクタ（つまり、Cisco Secure Firewall コネクタ）にエクスポートする方法のガイドラインです。詳細については、Cisco Secure Firewall ASA NetFlow 導入ガイド [英語] にある公式のシスコ コンフィギュレーション ガイドも参照してください。

次に、NSEL 設定の例を示します。

```
flow-export destination outside 172.29.142.27 4729
flow-export template timeout-rate 1
!
policy-map flow_export_policy
  class class-default
  flow-export event-type flow-create destination 172.29.142.27
  flow-export event-type flow-teardown destination 172.29.142.27
  flow-export event-type flow-denied destination 172.29.142.27
  flow-export event-type flow-update destination 172.29.142.27
  user-statistics accounting
service-policy flow_export_policy global
```

この例では、Cisco Secure Firewall ASA アプライアンスは、NetFlow パケットをポート 4729 の 172.29.142.27 に送信するように設定されています。さらに、flow-export アクションは、flow-create、flow-teardown、flow-denied、および flow-update イベントで有効になります。これらのフローイベントが ASA で発生すると、NetFlow レコードが生成され、設定で指定された宛先に送信されます。

Cisco Secure Firewall コネクタが、Secure Workload が有効になっていて、Secure Workload Ingest アプライアンスの 172.29.142.27:4729 でリッスンしていると仮定すると、コネクタは Cisco Secure Firewall ASA アプライアンスから NetFlow パケットを受信します。コネクタは、「NSEL イベントの処理」で説明されているように NetFlow レコードを処理し、フロー監視データを Cisco

Secure Workload にエクスポートしますさらに、NATed フローの場合、コネクタは関連するフロー（クライアント側とサーバー側）をステッチします。

コネクタの設定方法

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。コネクタでは、以下を設定できます。

- ログ：詳細については、「ログ設定」を参照してください。

さらに、許可されたコマンドを使用して、Secure Workload Ingest アプライアンスの Docker コンテナで、コネクタの IPFIX プロトコルのリスニングポートを更新できます。このコマンドは、コネクタのコネクタ ID、更新するポートのタイプ、および新しいポートの情報が提供されると、アプライアンスで発行できます。コネクタ ID は、Secure Workload UI の [コネクタ（connector）] ページにあります。詳細については、update-listening-ports を参照してください。

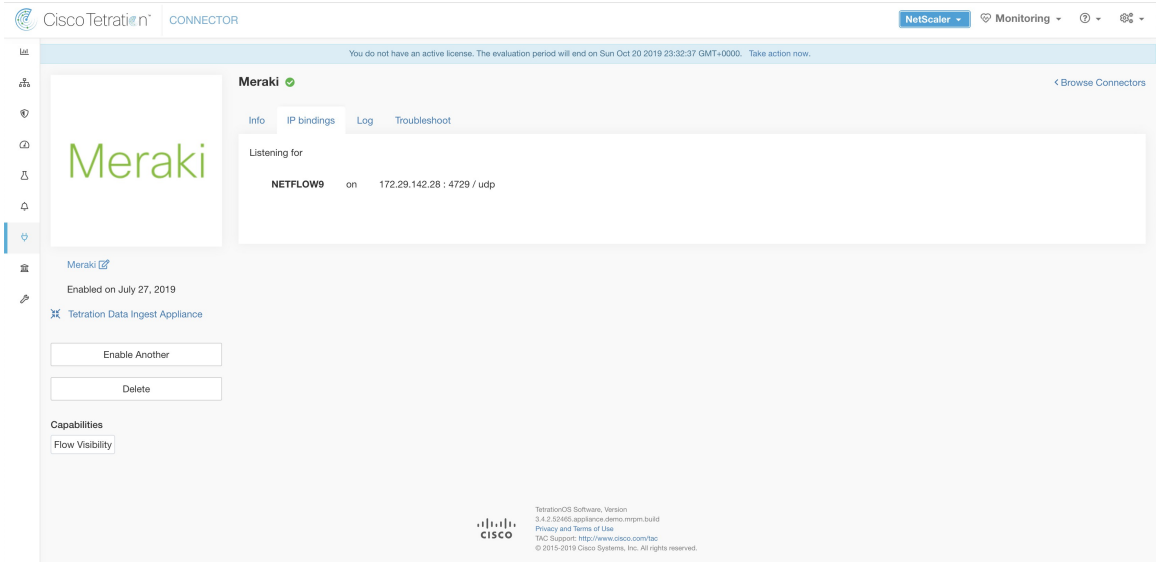
制限

メトリック	制限
1 つの Secure Workload Ingest アプライアンス上の Cisco Secure Firewall コネクタの最大数	1
1 つのテナント（ルート範囲）における Cisco Secure Firewall コネクタの最大数	10
Cisco Secure Workload における Cisco Secure Firewall コネクタの最大数	100

Meraki コネクタ

Meraki コネクタを使用すると、Secure Workload は Meraki ファイアウォール（Meraki MX セキュリティアプライアンスおよびワイヤレスアクセスポイントに含まれる）からフロー監視データを取り込むことができます。このソリューションを使用すると、Cisco スイッチにより、処理のために Secure Workload Ingest アプライアンスでホストされている Meraki コネクタに NetFlow レコードがリレーされるため、ホストでソフトウェアエージェントを実行する必要がありません。

図 *60 : Meraki* コネクタ



NetFlow とは

Netflow プロトコルを使用すると、Meraki ファイアウォールなどのネットワークデバイスは、デバイスを通過するトラフィックをフローに集約し、それらのフローをフローコレクタにエクスポートできます。フローコレクタはこれらのフローレコードを受け取ると、オフラインでクエリと分析を行えるようにフローストレージに保存します。

通常、セットアップには次の手順が含まれます。

1. Meraki ファイアウォールで NetFlow 統計レポートを有効にします。
2. Meraki ファイアウォールで NetFlow コレクタのエンドポイント情報を設定します。

Cisco Secure Workload へのフローの取り込み

Meraki コネクタは、本質的に NetFlow コレクタです。コネクタは、NetFlow トラフィック統計をエクスポートするように設定されている Meraki ファイアウォールから、フローレコードを受信します。NetFlow レコードを処理し、Meraki ファイアウォールによって報告されたフロー観測を、フロー分析のために Secure Workload に送信します。Meraki コネクタは、Ingest アプライアンスで有効にすることができ、Docker コンテナとして実行されます。Secure Workload

Meraki コネクタは、Secure Workload Meraki エージェントとしても Secure Workload で登録されます。Meraki コネクタは、NetFlow プロトコルパケット（フローレコード）のカプセル化を解除します。次に、通常の Secure Workload エージェントのようにフローを処理してレポートします。Deep Visibility Agent とは異なり、プロセスやインターフェースの情報は報告しません。

（注） Meraki コネクタは NetFlow v9 プロトコルをサポートしています。



（注） 各 Meraki コネクタは、1 つの VRF のフローのみを報告する必要があります。コネクタによってエクスポートされたフローは、Secure Workload クラスタのエージェント VRF 設定に基づいて VRF に配置されます。コネクタの VRF を設定するには、**[管理（Manage）]** > **[エージェント（Agents）]**に移動し、[設定（Configuration）] タブをクリックします。このページの [エージェントのリモートVRF設定（Agent Remote VRF Configurations）] セクションで、[設定の作成（Create Config）] をクリックし、コネクタに関する詳細を指定します。このフォームで、VRF の名前、コネクタの IP サブネット、およびフローレコードをクラスタに送信できる可能性のあるポート番号の範囲を提供するようにユーザーに要求します。

NetFlow レコードの処理

Meraki コネクタは、NetFlow レコードに基づいてフロー観察情報を Cisco Secure Workload に送信します。Meraki NetFlow フローレコードは双方向レコードです。したがって、Meraki コネクタは、Cisco Secure Workload への順方向フローと逆方向フローの 2 つのフローを送信します。

以下は、Meraki コネクタから Cisco Secure Workload に送信されるフロー観察情報の詳細です。

フォワードフロー観測データ

フィールド	Element ID	エレメント名
[プロトコル（Protocol）]	4	*protocolIdentifier*
[送信元アドレス（Source Address）]	8	*sourceIPv4Address*
[送信元ポート（Source Port）]	7	*sourceTransportPort*
[宛先アドレス（Destination Address）]	12	*destinationIPv4Address*
[宛先ポート（Destination Port）]	11	*destinationTransportPort*
Byte Count	1	*octetDeltaCount*
Packet Count	2	*packetDeltaCount*
フロー開始時刻		このフローの NetFlow レコードがコネクタでいつ受信されたかに基づいて設定されます

リバースフロー観測情報

フィールド	Element ID	
[プロトコル（Protocol）]	4	*protocolIdentifier*

フィールド	Element ID	
[送信元アドレス（Source Address）]	8	*sourceIPv4Address*
[送信元ポート（Source Port）]	7	*sourceTransportPort*
[宛先アドレス（Destination Address）]	12	*destinationIPv4Address*
[宛先ポート（Destination Port）]	11	*destinationTransportPort*
Byte Count	23	*postOctetDeltaCount*
Packet Count	24	*postPacketDeltaCount*
フロー開始時刻		このフローの NetFlow レコードが コネクタでいつ受信されたかに基づいて設定されます

Meraki ファイアウォールでの NetFlow の設定方法

次の手順は、Meraki ファイアウォールで NetFlow レポートを設定する方法を示しています。

手順

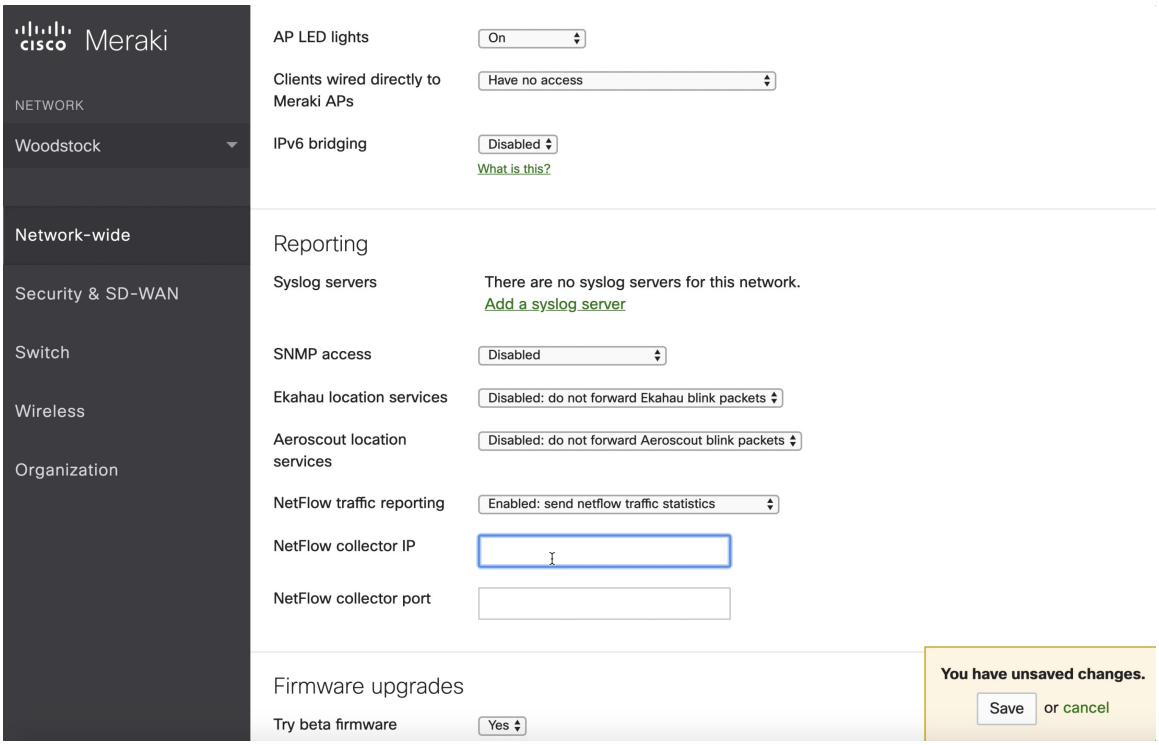
ステップ 1 Meraki UI コンソールにログインします。

ステップ 2 **[ネットワーク全体（Network-wide）]**>**[全般（General）]**に移動します。[レポート（Reporting）] の設定で、[NetFlowトラフィックレポート（NetFlow traffic reporting）] を有効にします。値が [有効: NetFlowトラフィック統計情報の送信（Enabled: send NetFlow traffic statistics）] になっていることを確認します。

ステップ 3 Meraki コネクタが Secure Workload Ingest アプライアンスでリッスンしている IP およびポートを [NetFlowコレクタIP（NetFlow collector IP）] および [NetFlowコレクタポート（NetFlow collector port）] で指定します。Meraki コネクタが NetFlow レコードをリッスンするデフォルトのポートは 4729 です。

ステップ 4 変更内容を保存します。

図 *61 : Meraki* ファイアウォールでの *NetFlow* の有効化



コネクタの設定方法

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。コネクタでは、以下を設定できます。

- ログ：詳細については、「ログ設定」を参照してください。

さらに、許可されたコマンドを使用して、Secure Workload Ingest アプライアンスの Docker コンテナで、コネクタの NetFlow v9 プロトコルのリスニングポートを更新できます。このコマンドは、コネクタのコネクタ ID、更新するポートのタイプ、および新しいポートの情報が提供されると、アプライアンスで発行できます。コネクタ ID は、Secure Workload UI の [コネクタ（connector）] ページにあります。詳細については、update-listening-ports を参照してください。

制限

メトリック	制限
1 つの Secure Workload Ingest アプライアンスにおける Meraki コネクタの最大数	1
1 つのテナント（ルート範囲）における Meraki コネクタの最大数	10

メトリック	制限
Cisco Secure Workload における Meraki コネクタの最大数	100

ERSPAN コネクタ

ERSPAN コネクタを使用することにより、Secure Workload は、ネットワーク内のルータおよびスイッチからフロー観測データを取り込むことができます。このソリューションを使用すると、Cisco スイッチがホストのトラフィックを ERSPAN コネクタに中継して処理するため、ホストはソフトウェアエージェントを実行する必要がありません。

ERSPAN について

Encapsulated Remote Switch Port Analyzer（ERSPAN）は、ほとんどの Cisco スイッチに存在する機能です。ERSPAN は、ネットワークデバイスによって認識されるフレームをミラーリングし、IP パケットにカプセル化して、リモートアナライザに送信します。ユーザーは、監視するスイッチ上のインターフェイスや VLAN のリストを選択できます。

一般に、セットアップでは、1 つ以上のネットワークデバイスで送信元 ERSPAN モニタリングセッションを設定し、トラフィックアナライザに直接接続されているリモート ネットワークデバイスで宛先 ERSPAN モニタリングセッションを設定します。

Secure Workload ERSPAN コネクタでは、宛先 ERSPAN セッションとトラフィックアナライザ機能の両方が提供されるため、Secure Workload ソリューションを使用してスイッチで宛先セッションを構成する必要はありません。

SPAN エージェントとは

各 ERSPAN コネクタは、クラスタに SPAN エージェントを登録します。Secure Workload SPAN エージェントは、ERSPAN パケットのみを処理するように構成された通常の Secure Workload エージェントです。シスコの宛先 ERSPAN セッションと同様に、ミラーリングされたフレームのカプセル化を解除します。その後、通常の Secure Workload エージェントのようにフローを処理して報告します。優れた可視性エージェントとは異なり、プロセスやインターフェイスの情報を報告しません。

ERSPAN のIngest アプライアンスとは

Secure WorkloadERSPAN の Ingest アプライアンスは、3 つの ERSPAN Secure Workloadコネクタを内部で実行する VM です。通常の Ingest アプライアンスと同じ OVA または QCOW2 を使用します。

各コネクタは、1 つの vNIC と 2 つの vCPU コア（制限クォータなし）が排他的に割り当てられた専用の Docker コンテナ内で実行されます。

ERSPAN コネクタは、コンテナホスト名 <VM hostname>-<interface IP address> を持つクラスタに ERSPAN エージェントを登録します。

コネクタとエージェントは、VM、Docker デーモン、または Docker コンテナのクラッシュまたは再起動時に、保持または復元されます。

（注） ERSPAN コネクタのステータスは、[コネクタ（Connector）]ページに報告されます。[エージェントリスト（Agent List）]ページを参照して、対応する SPAN エージェントの状態を確認してください。

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。ERSPAN コネクタの場合、IPv4 および IPv6（デュアルスタックモード）アドレスがサポートされます。ただし、デュアルスタックのサポートはベータ版の機能であることに注意してください。

送信元 ERSPAN セッションの構成方法

次の手順は、Nexus 9000 スイッチ用です。他のシスコプラットフォームでは、設定方法が若干異なる場合があります。いずれの場合も、設定するシスコプラットフォームに関する公式のシスコ コンフィグレーション ガイドも参照してください。

図 *62 : Cisco Nexus 9000* での *ERSPAN* 送信元の構成

```
Enter the configuration mode
# config terminal

Configure the erspan source IP address
(config)# monitor erspan origin ip-address 172.28.126.1 global

Create and configure the source erspan session
(config)# monitor session 10 type erspan-source
(config-erspan-src)# source interface ethernet 1/23 both
(config-erspan-src)# source vlan 315, 512
(config-erspan-src)# destination ip 172.28.126.194

Turn on the monitor session
(config-erspan-src)# no shut

Persist the configuration
# copy runnin-config startup-confi
```

上記の手順により、ID 10 の送信元 ERSPAN セッションが作成されました。スイッチは、インターフェイス eth1/23 の入力と出力（both）フレーム、および VLAN 315 と 512 上のフレームをミラーリングします。ミラーリングされたフレームを運ぶ外部 GRE パケットには、送信元 IP 172.28.126.1（このスイッチの L3 インターフェイスのアドレス）と宛先 IP 172.28.126.194 が含まれます。これは、ERSPAN VM で構成されている IP アドレスの 1 つです。

サポートされている ERSPAN 形式

Secure Workload SPAN エージェントは、提案されている ERSPAN RFC で説明した ERSPAN タイプ I、II、および III パケットを処理できます。したがって、Cisco デバイスによって生成された ERSPAN パケットは処理可能です。RFC に準拠していない形式の中では、VMware vSphere Distributed Switch（VDS）によって生成された ERSPAN パケットを処理できます。

ERSPAN 送信元を設定する際のパフォーマンス上の考慮事項

ERSPAN 送信元のポート/VLAN リストを慎重に選択します。SPAN エージェントには 2 つの専用 vCPU がありますが、セッションによって大量のパケットが生成され、エージェントの処理能力が飽和する可能性があります。エージェントが処理能力を超えるパケットを受信している場合、クラスタの [優れた可視性エージェント（Deep Visibility Agent）] ページの [エージェントパケット欠落（Agent Packet Misses）] グラフに表示されます。

ERSPAN 送信元がミラーリングするフレームの詳細な調整は、通常は filter 構成キーワードを指定し、ACL ポリシーを使用して行います。

スイッチでサポートされている場合、通常は mtu キーワードを使用して、ERSPAN パケットの最大伝送ユニット（MTU）を変更するように ERSPAN 送信元セッションを設定できます（通常、デフォルト値は 1500 バイト）。MTU 値を小さくすると、ネットワーク インフラストラクチャでの ERSPAN 帯域幅の使用が制限されますが、エージェントのワークロードはパケット単位であるため、SPAN エージェントの負荷には影響しません。MTU 値を小さくする場合は、ミラーフレーム用に 160 バイトのスペースを確保してください。ERSPAN ヘッダーオーバーヘッドの詳細については、提案されている ERSPAN RFC を参照してください。

ERSPAN には 3 つのバージョンがあります。バージョンが小さいほど、ERSPAN ヘッダーのオーバーヘッドは低くなります。バージョン II および III では、ERSPAN パケットに QoS ポリシーを適用し、複数の VLAN 情報を提供できます。バージョン III には、さらに多くの設定が含まれています。バージョン II は通常、Cisco スイッチのデフォルトです。Secure Workload SPAN エージェントは 3 つのバージョンをすべてサポートしていますが、現時点では、ERSPAN バージョン II および III パケットに含まれる追加情報は利用していません。

セキュリティの考慮事項

ERSPAN ゲスト オペレーティング システムの取り込み仮想マシンは CentOS 7.9 であり、そこから OpenSSL サーバー/クライアントパッケージが削除されています。

VM が起動し、SPAN エージェント コンテナが展開されると（初回起動時のみ数分かかります）、ループバック以外のネットワーク インターフェイスは仮想マシンに表示されません。そのため、アプライアンスにアクセスするにはそのコンソールを使用するしか方法がありません。

VM ネットワーク インターフェイスは Docker コンテナ内に移動しました。コンテナは、TCP/UDP ポートを開かずに、centos:7.9.2009 ベースの Docker イメージを実行します。

また、コンテナは基本権限（–privileged オプションなし）に加え、NET_ADMIN 機能を使用して実行されます。

万が一、コンテナが侵害された場合でも、VM ゲスト OS はコンテナの内部から侵害されないようにするべきです。

ホスト内で実行される Secure Workload エージェントで有効な他のすべてのセキュリティ上の考慮事項は、Docker コンテナ内で実行される Secure Workload SPAN エージェントにも適用されます。

トラブルシューティング

クラスタの [モニタリング/エージェントの概要（Monitoring/Agent Overview）] ページに SPAN エージェントがアクティブな状態で表示された後は、ERSPAN 仮想マシンで操作する必要はなく、ユーザーがログインする必要もありません。エージェントが表示されない場合や、フローがクラスタに報告されない場合は、次の情報が展開の問題を特定するのに役立ちます。

通常の状態では、VM で次の情報を表示できます。

- `systemctl status tet_vm_setup` により、*SUCCESS* 終了ステータスになっている非アクティブなサービスがレポートされます。
- `systemctl status tet-nic-driver` により、アクティブなサービスがレポートされます。
- `docker network ls` により、5 つのネットワーク（`host`、`none` および 3 つの `erspan-<iface name>`）がレポートされます。
- `ip link` により、ループバック インターフェイスのみがレポートされます。
- `docker ps` により、3 つの実行中のコンテナがレポートされます。
- 各コンテナの `docker logs <cid>` には次のメッセージが含まれます。`INFO success: tet-sensor entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)`
- `docker exec <cid> ifconfig` により、ループバックに加えて 1 つのインターフェイスのみがレポートされます。
- `docker exec <cid> route -n` により、デフォルトゲートウェイがレポートされます。
- `docker exec <cid> iptables -t raw -S PREROUTING` により、ルール `-A PREROUTING -p gre -j DROP` がレポートされます。

上記のいずれにも当てはまらない場合は、`/local/tetration/logs/tet_vm_setup.log` の展開スクリプトログで、SPAN エージェントコンテナの展開が失敗した理由を確認してください。

他のエージェントの登録/接続の問題は、*docker exec* コマンドを使用して、ホストで実行されているエージェントと同じ方法でトラブルシューティングできます。

- `docker exec <cid> ps -ef` により、`tet-engine, tet-engine check_conf` の 2 インスタンス、および `/usr/local/tet/tet-sensor -f /usr/local/tet/conf/.sensor_config` の 2 インスタンスがレポートされます。1 つは root ユーザー、もう 1 つは tet-sensor ユーザーに関するものです。加えて、プロセスマネージャ `/usr/bin/ python /usr/bin/supervisord -c /etc/supervisord.conf -n` インスタンスもレポートされます。
- `docker exec <cid> cat /usr/local/tet/log/tet-sensor.log` により、エージェントのログが表示されます。
- `docker exec <cid> cat /usr/local/tet/log/fetch_sensor_id.log` により、エージェントの登録ログが表示されます。
- `docker exec <cid> cat /usr/local/tet/log/check_conf_update.log` により、設定更新ポーリングログが表示されます。

必要に応じて、tcpdump をコンテナのネットワーク名前空間に設定し、コンテナとの間のトラフィックをモニターできます。

1. docker inspect <cid> | grep SandboxKey を使用して、コンテナのネットワーク名前空間（SandboxKey）を取得します。
2. コンテナのネットワーク名前空間 nsenter --net=/var/run/docker/netns/... に対して設定します。
3. eth0 traffic tcpdump -i eth0 -n をモニターします。

制限

メトリック	制限
1 つの Secure Workload Ingest アプライアンスにおける ERSPAN コネクタの最大数	3
1 つのテナント（ルート範囲）における ERSPAN コネクタの最大数	24（TaaS の場合は 12）
Cisco Secure Workload における ERSPAN コネクタの最大数	450

エンドポイントのコネクタ

エンドポイントのコネクタは、Cisco Secure Workload のエンドポイントコンテキストを提供します。

コネクタ	説明	仮想アプライアンス上に展開
AnyConnect	Cisco AnyConnect Network Visibility Module（NVM）からテレメトリデータを収集し、エンドポイントインベントリをユーザー属性で強化します。	Cisco Secure Workload Ingest
ISE	Cisco ISE アプライアンスによって管理されるエンドポイントとインベントリに関する情報を収集し、エンドポイントインベントリをユーザー属性とセキュアグループラベル（SGL）で強化します。	Cisco Secure Workload Edge

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。

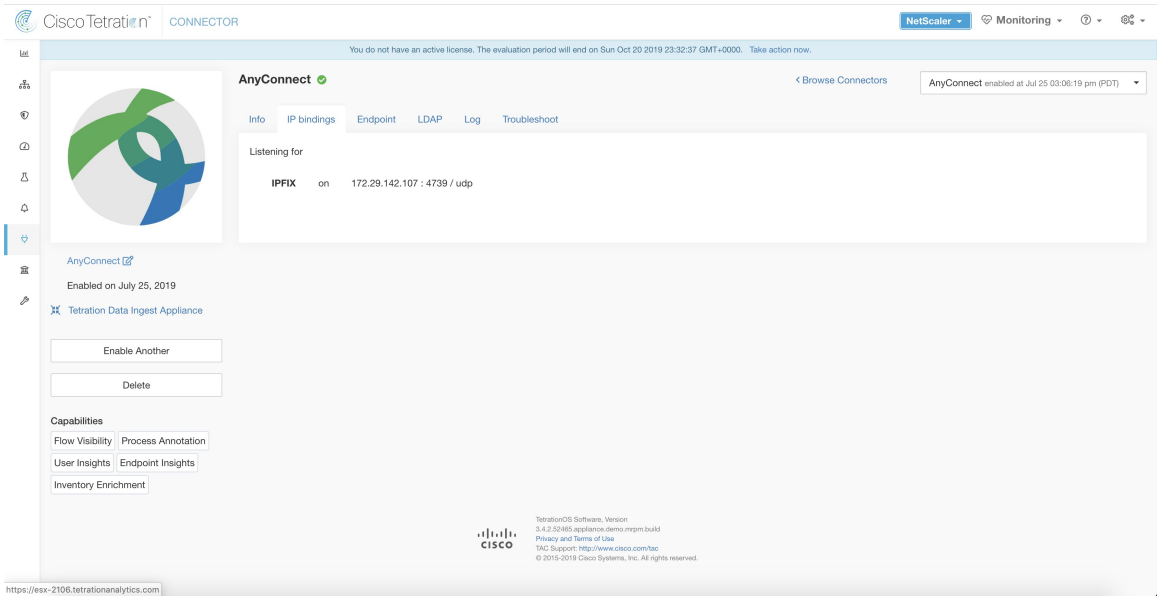
AnyConnect コネクタ

AnyConnect コネクタは、ネットワーク可視性モジュール（NVM）を備えた Cisco AnyConnect セキュアモビリティクライアントを実行するエンドポイントを監視します。このソリューションを使用すると、NVM がホスト、インターフェイス、フローレコードを IPFIX 形式で AnyConnect コネクタなどのコレクタに送信するため、ホストはエンドポイントでソフトウェアエージェントを実行する必要がありません。

AnyConnect コネクタは、次の高度な機能を実行します。

1. 各エンドポイント（デスクトップ、ラップトップ、スマートフォンなどのサポートされているユーザーデバイス）を、Cisco Secure Workload で AnyConnect エージェントとして登録します。
2. Cisco Secure Workload を使用して、登録されたエンドポイントからインターフェイスのスナップショットを更新します。
3. 登録されたエンドポイントによってエクスポートされたフロー情報を Secure Workload コレクタに送信します。
4. AnyConnect コネクタによって追跡されるエンドポイントでフローを生成するプロセスの、プロセススナップショットを定期的に送信します。
5. 各エンドポイントでログインしているユーザーに対応する Lightweight Directory Access Protocol （LDAP）属性を使用して、エンドポイント インターフェイスの IP アドレスにラベルを付けます。

図 *63 : AnyConnect* コネクタ



AnyConnect NVM について

AnyConnect NVM は、オンプレミスとオフプレミスの両方でエンドポイントとユーザーの動作を可視化およびモニタリングします。次のコンテキストを含むエンドポイントから情報を収集します。

1. **デバイス/エンドポイントコンテキスト**：デバイス/エンドポイント固有の情報。
2. **ユーザーコンテキスト**：フローに関連付けられたユーザー。
3. **アプリケーション コンテキスト**：フローに関連付けられたプロセス。
4. **ロケーションコンテキスト**：ロケーション固有の属性（利用可能な場合）。
5. **接続先コンテキスト**：宛先の FQDN。AnyConnect NVM は 3 つのタイプのレコードを生成します。

NVM レコード	説明
エンドポイントレコード	一意のデバイス識別子（UDID）、ホスト名、OS 名、OS バージョン、製造元などのデバイス/エンドポイント情報。
インターフェイスレコード	エンドポイント UDID、インターフェイス固有識別子（UID）、インターフェイス インデックス、インターフェイスタイプ、インターフェイス名、MACアドレスなど、エンドポイントの各インターフェイスに関する情報。
Flow Record	エンドポイント UDID、インターフェイス UID、5 タプル（送信元/宛先 IP/ポートおよびプロトコル）、イン/アウトバイトカウント、プロセス情報、ユーザー情報、宛先の FQDN など、エンドポイントで観測されるフローについての情報。

各レコードは、IPFIXプロトコル形式で生成およびエクスポートされます。デバイスが信頼ネットワーク（オンプレミス/VPN）にある場合、AnyConnect NVM は設定されたコレクタにレコードをエクスポートします。AnyConnect コネクタは、AnyConnect NVM から IPFIX ストリームを受信して処理できる IPFIX コレクタの一例です。



（注） AnyConnect コネクタは、Cisco AnyConnect セキュア モビリティ クライアントのバージョン 4.2 以降の AnyConnect NVM をサポートします。

AnyConnect NVM の設定方法

Cisco Secure Firewall ASA または Cisco Identity Services Engine (ISE) を使用して AnyConnect NVM を導入する手順については、『How to Implement AnyConnect NVM』[英語] を参照してください。NVM モジュールが展開されたら、NVM プロファイルを指定し、Cisco AnyConnect セキュア モビリティ クライアントを稼働中のエンドポイントにプッシュしてインストールする必要があります。NVM プロファイルを指定する際、ポート 4739 の AnyConnect コネクタを指すように IPFIX コレクタを設定する必要があります。

AnyConnect コネクタは、Secure Workload AnyConnect プロキシ エージェントとして Secure Workload にも登録されます。

NVM レコードの処理

AnyConnect コネクタは、次に示すように AnyConnect NVM レコードを処理します。

エンドポイントレコード

AnyConnect コネクタがエンドポイントレコードを受信すると、そのエンドポイントは Cisco Secure Workload 上の AnyConnect エージェントとして登録されます。AnyConnect コネクタは、NVM レコードに存在するエンドポイント固有の情報と AnyConnect コネクタの証明書を使用して、エンドポイントを登録します。エンドポイントが登録されると、Cisco Secure Workload 内のいずれかのコレクタへの新しい接続を作成することにより、エンドポイントのデータプレーンが有効になります。このエンドポイントからのアクティビティ（フローレコード）に基づいて、AnyConnect コネクタは、クラスタでこのエンドポイントに対応する AnyConnect エージェントを定期的に（20 ～ 30 分）チェックインします。

AnyConnect NVM は、4.9 以降のエージェントバージョンで送信を開始します。デフォルトでは、AnyConnect エンドポイントは Cisco Secure Workload 上でバージョン 4.2.x として登録されます。このバージョンは、サポートされる AnyConnect NVM の最小バージョンを意味します。バージョン 4.9 以降の AnyConnect エンドポイントの場合、Secure Workload 上の対応する AnyConnect エージェントには、インストールされている実際のバージョンが示されます。



（注） AnyConnect エージェントのインストールバージョンは、Cisco Secure Workload によって制御されません。Secure Workload UI で AnyConnect エンドポイントエージェントをアップグレードしようとしても、効果がありません。

インターフェイスレコード

インターフェイスのインターフェイスレコード IP アドレスは、AnyConnect NVM インターフェイスレコードの一部ではありません。インターフェイスの IP アドレスは、そのインターフェイスのエンドポイントからフローレコードが送信され始めるときに決定されます。インターフェイスの IP アドレスが決定されると、AnyConnect コネクタは、IP アドレスが決定されたそのエンドポイントのすべてのインターフェイスの完全なスナップショットを Cisco Secure Workload の構成サーバーに送信します。これにより、VRF がインターフェイスデータに関連付けられ、これらのインターフェイスに着信するフローがこの VRF でマークされるようになります。

Flow Record

フローレコードを受信すると、AnyConnect コネクタはそのレコードを Secure Workload が認識できる形式に変換し、そのエンドポイントに対応するデータプレーンを介して FlowInfo を送信します。さらに、フローレコードに含まれるプロセス情報をローカルに保存します。また、AnyConnect コネクタに LDAP 設定が提供されている場合、エンドポイントのログインユーザーの設定済み LDAP 属性の値が決定されます。属性は、フローが発生したエンドポイントの IP アドレスに関連付けられています。定期的に、プロセス情報とユーザーラベルが Cisco Secure Workload にプッシュされます。

（注） 各 AnyConnect コネクタは、1 つの VRF のみのエンドポイント/インターフェイス/フローを報告します。AnyConnect コネクタによって報告されるエンドポイントとインターフェイスは、Cisco Secure Workload のエージェント VRF コンフィギュレーションに基づいて VRF に関連付けられます。AnyConnect エンドポイントの代わりに AnyConnect コネクタエージェントによってエクスポートされたフローは、同じ VRF に属します。エージェントの VRF を設定するには、**[管理（Manage）]** > **[エージェント（Agents）]** に移動し、[構成（Configuration）] タブをクリックします。このページの [エージェントのリモートVRF構成（Agent Remote VRF Configurations）] セクションで、[構成の作成（Create Config）] をクリックし、AnyConnect コネクタに関する詳細を指定します。このフォームでは、VRF の名前、エージェントがインストールされているホストの IP サブネット、およびフローレコードをクラスタに送信する可能性のあるポート番号の範囲を提供するようにユーザーに要求します。

Windows エンドポイントでの UDID の重複

エンドポイントマシンが同じゴールデンイメージから複製された場合、複製されたすべてのエンドポイントの UDID が同一である可能性があります。同一の場合、AnyConnect コネクタは、同一の UDID を持つエンドポイントからエンドポイントレコードを受信し、同じ UDID で Secure Workload に登録します。コネクタは、それらのエンドポイントからインターフェイスまたはフローレコードを受信すると、データを関連付けるために Secure Workload の正しい AnyConnect エージェントを判別できないため、すべてのデータを 1 つのエンドポイントに関連付けます（確定的ではありません）。

この問題に対処するために、AnyConnect NVM 4.8 リリースには、エンドポイントで UDID を見つけて再生成する dartcli.exe というツールが同梱されています。

- dartcli.exe -u は、エンドポイントの UDID を取得します。
- dartcli.exe -nu は、エンドポイントの UDID を再生成します。このツールを実行するには、次の手順を使用してください。

```
  C:\Program Files (x86)\Cisco\Cisco AnyConnect Secure Mobility Client\DART>dartcli.exe
-u
   UDID : 8D0D1E8FA0AB09BE82599F10068593E41EF1BFFF

  C:\Program Files (x86)\Cisco\Cisco AnyConnect Secure Mobility Client\DART>dartcli.exe
-nu
   Are you sure you want to re-generate UDID [y/n]: y
   Adding nonce success
```

```
    UDID : 29F596758941E606BD0AFF49049216ED5BB9F7A5

   C:\Program Files (x86)\Cisco\Cisco AnyConnect Secure Mobility Client\DART>dartcli.exe
 -u
    UDID : 29F596758941E606BD0AFF49049216ED5BB9F7A5
```

定期的タスク

AnyConnect コネクタは、AnyConnect エンドポイントインベントリでプロセススナップショットとユーザーラベルを定期的に送信します。

1. [プロセススナップショット（Process Snapshot）]：AnyConnect コネクタは、5 分ごとに、その間隔でローカルで保持されているプロセスを調べ、その間隔中にフローがあったすべてのエンドポイントのプロセススナップショットを送信します。
2. [ユーザーラベル（User Labels）]：AnyConnect コネクタは、2 分ごとに、ローカルで保持されている LDAP ユーザーラベルを調べ、それらの IP アドレスのユーザーラベルを更新します。

ユーザーラベル用に、AnyConnect コネクタは、組織に属する全ユーザーの LDAP 属性のローカルスナップショットを作成します。AnyConnect コネクタが有効になっている場合、LDAP の設定（サーバー/ポート情報、ユーザーに関して取得される属性、ユーザー名を含む属性）が提供される場合があります。さらに、LDAP サーバーにアクセスするための LDAP ユーザーログイン情報が提供されることもあります。LDAP ユーザーログイン情報は暗号化され、AnyConnect コネクタで公開されることはありません。オプションで、LDAP サーバーに安全にアクセスするための LDAP 証明書の提供も可能です。

（注） AnyConnect コネクタは、24 時間ごとに新しいローカル LDAP スナップショットを作成します。この間隔は、コネクタの LDAP 設定で変更できます。

コネクタの設定方法

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。コネクタでは、以下を設定できます。

- *LDAP*：LDAP 設定は、LDAP 属性の検出をサポートし、ユーザー名に対応する属性を選択するワークフローと、ユーザーごとに取得される最大 6 つの属性のリストを提供します。詳細については、「検出」を参照してください。
- エンドポイント：詳細については、「エンドポイントの設定」を参照してください。
- ログ：詳細については、「ログ設定」を参照してください。

さらに、許可されたコマンドを使用して、Secure Workload Ingest アプライアンスの Docker コンテナで、コネクタの IPFIX プロトコルのリスニングポートを更新できます。このコマンドは、コネクタのコネクタ ID、更新するポートのタイプ、および新しいポートの情報が提供されると、アプライアンスで発行できます。コネクタ ID は、Secure Workload UI の [コネクタ（connector）]ページにあります。詳細については、update-listening-ports を参照してください。

制限

メトリック	制限
1 つの Secure Workload Ingest アプライアンスにおける AnyConnect コネクタの最大数	1
1 つのテナント（ルート範囲）における AnyConnect コネクタの最大数	50
Cisco Secure Workload における AnyConnect コネクタの最大数	500

ISE コネクタ

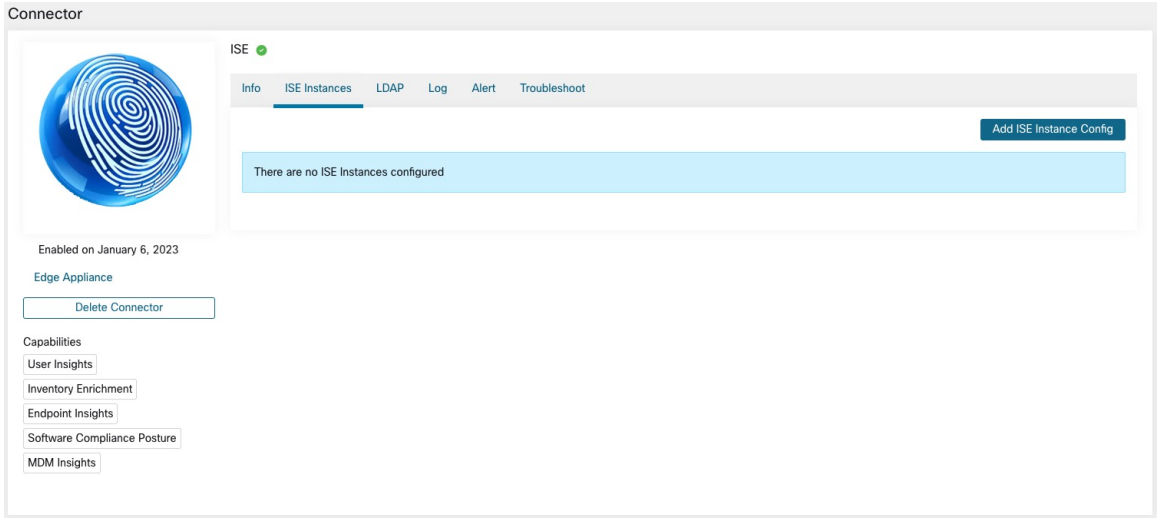
Cisco Secure Workload の ISE コネクタは、Cisco Platform Exchange Grid（pxGrid）を使用して Cisco Identity Services Engine（ISE）に接続し、ISE によって報告されるエンドポイントに関するコンテキスト情報を取得します。このソリューションを使用すると、エンドポイントの豊富なメタデータを取得できます。

Cisco Secure Workload の ISE コネクタは、Cisco Platform Exchange Grid（pxGrid）を使用して Cisco Identity Services Engine（ISE）と ISE Passive Identity Connector（ISE-PIC）に接続し、ISE によって報告されるエンドポイントのコンテキスト情報（メタデータなど）を取得します。

ISE コネクタは、次の機能を実行します。

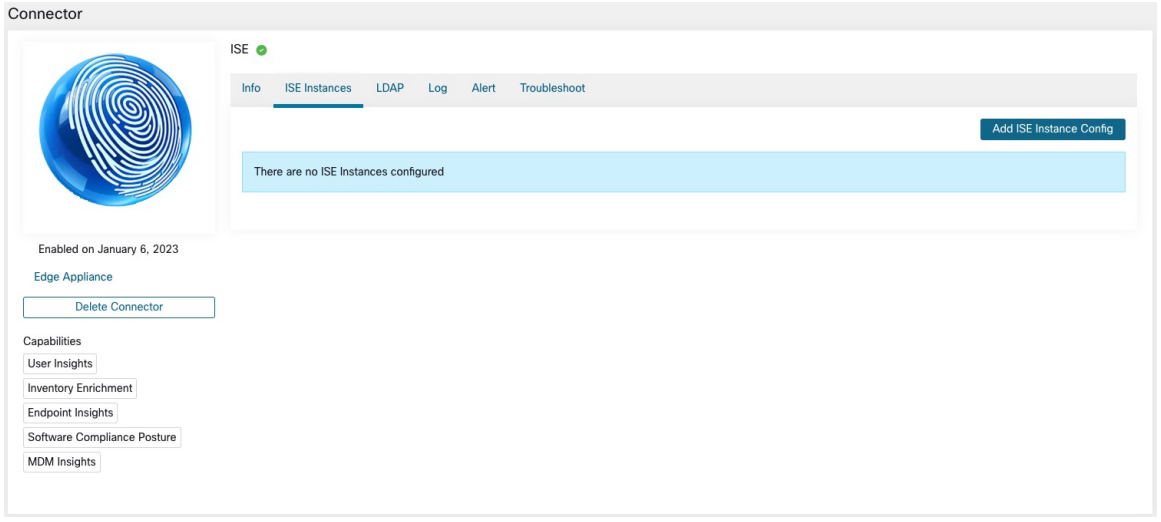
1. ISE によって表示される各エンドポイントを、Secure Workload で ISE エンドポイントエージェントとして登録します。
2. Secure Workload へのこれらのエンドポイントに関するメタデータ情報（MDM の詳細、認証、セキュリティグループラベルなどを含む）を更新します。
3. 定期的にスナップショットを作成し、ISE で表示されるアクティブなエンドポイントでクラスタを更新します。

図 *64 : ISE* コネクタ



1. Cisco Secure Workload の ISE エンドポイントとして識別された各エンドポイントを登録します。
2. MDM の詳細、認証、セキュリティグループラベル、ISE グループ名、および ISE グループタイプなど、エンドポイントに関する Cisco Secure Workload のメタデータ情報を更新します。
3. 定期的にスナップショットを作成し、ISE で表示されるアクティブなエンドポイントでクラスタを更新します。

図 *65 : ISE* コネクタ



（注） 各 ISE コネクタは、1 つの VRF のエンドポイントとインターフェイスのみを登録します。ISE コネクタによって報告されるエンドポイントとインターフェイスは、Cisco Secure Workload のエージェント VRF 設定に基づき、VRF に関連付けられます。エージェントの VRF を設定するには、**[管理（Manage）]** > **[エージェント（Agents）]** に移動し、[構成（Configuration）] タブをクリックします。このページの [エージェントのリモートVRF設定（Agent Remote VRF Configurations）] セクションで、[設定の作成（Create Config）] をクリックし、ISE コネクタに関する詳細を指定します。このフォームでは、VRF の名前、エージェントがインストールされているホストの IP サブネット、および Cisco Secure Workload の ISE エンドポイントおよびインターフェイスを登録する可能性があるポート番号の範囲を提供するようにユーザーに要求します。

（注） ISE エンドポイントエージェントは、[エージェントリスト（Agent List）] ページには表示されません。代わりに、属性を持つ ISE エンドポイントを [インベントリ（Inventory）] ページで表示できます。

コネクタの設定方法

（注） この統合には、ISE バージョン 2.4+ と ISE PIC バージョン 3.1+ が必要です。

この統合には、ISE バージョン 2.4+ が必要です。

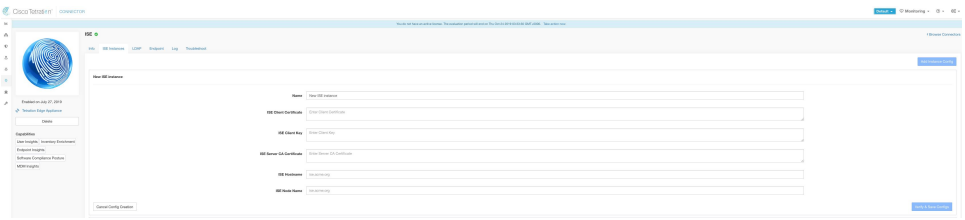
必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。ISE コネクタの場合、IPv4 および IPv6（デュアルスタックモード）アドレスがサポートされます。ただし、デュアルスタックのサポートはベータ機能であることに注意してください。

コネクタでは、次の設定が許可されています。

- ISE インスタンス：ISE コネクタは、指定された設定を使用して ISE の複数のインスタンスに接続できます。各インスタンスには、ISE に接続するためのホスト名およびノード名とともに ISE 証明書のログイン情報が必要です。詳細については、「ISE インスタンスの設定」を参照してください。
- LDAP：LDAP 設定は、LDAP 属性の検出をサポートし、ユーザー名に対応する属性を選択するワークフローと、ユーザーごとに取得される最大 6 つの属性のリストを提供します。詳細については、「検出」を参照してください。
- エンドポイント：詳細については、「エンドポイントの設定」を参照してください。
- ログ：詳細については、「エンドポイントの設定」を参照してください。

ISE インスタンスの設定

図 *66 : ISE* インスタンスの設定



（注） Cisco Secure Workload バージョン 3.7 以降、Cisco ISE pxGrid ノードの SSL 証明書には、この統合のためにサブジェクト代替名（SAN）が必要になります。Cisco Secure Workload との統合を実行する前に、ISE 管理者が ISE ノードの認証設定を行っていることを確認してください。

pxGrid ノードの証明書を確認し、SAN が設定されているかどうかを確認するには、次の手順を実行して ISE からの証明書を確認する必要があります。

手順

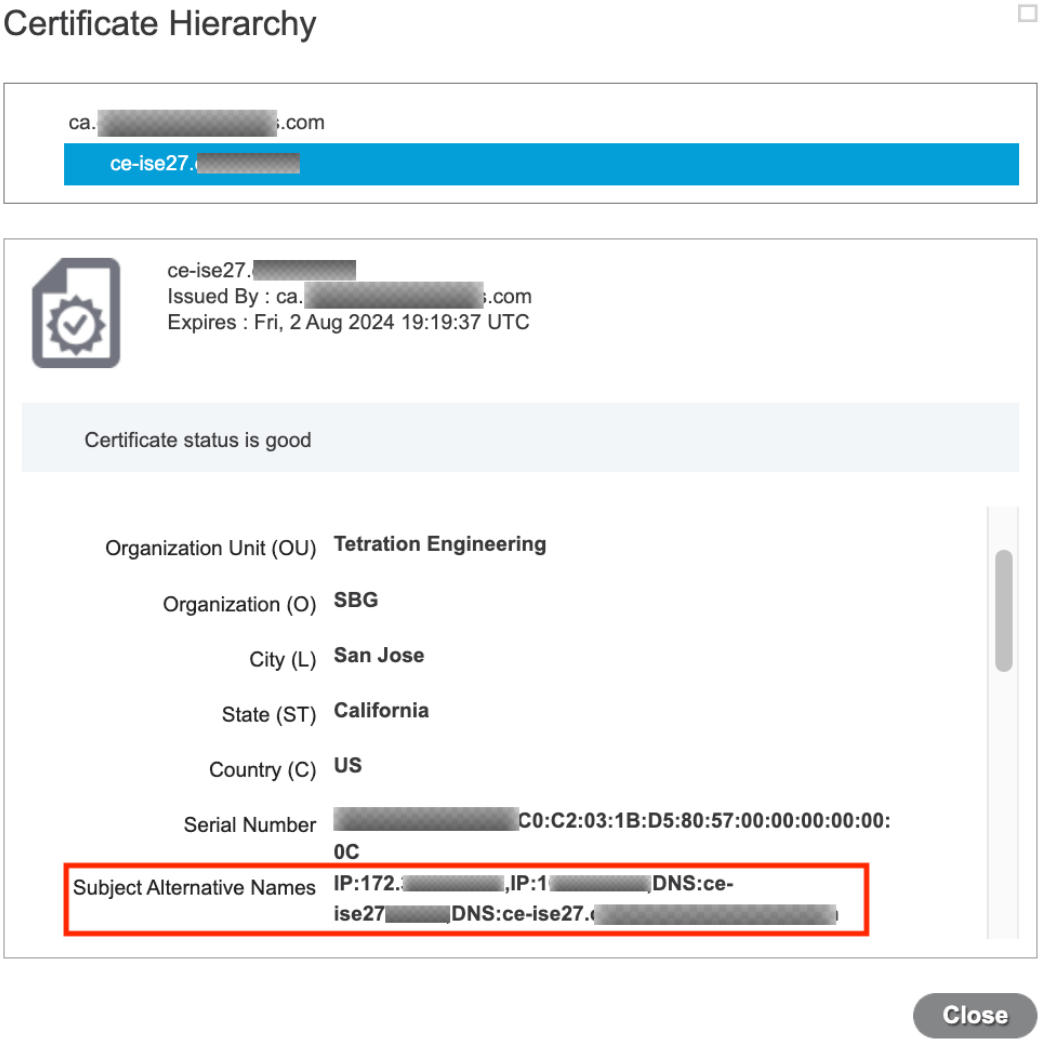
ステップ 1 **[管理（Administration）] > [システム（System）]** から [証明書（Certificates）] に移動します。

ステップ 2 [証明書の管理（Certificate Management）] で、[システム証明書（System Certificates）] を選択し、「使用される」pxGrid 証明書を選択し、[表示（View）] を選択して pxGrid ノード証明書を確認します。

ステップ 3 証明書をスクロールし、サブジェクト代替名がこの証明書に設定されていることを確認します。

ステップ 4 この証明書は、有効な認証局（CA）によって署名されている必要があり、これは、Cisco Secure Workload ISE コネクタに使用される pxGrid クライアント証明書の署名にも使用されている必要があります。

図 *67 :* 有効な *ISE pxGrid* ノード証明書の例



ステップ 5 OpenSSL がインストールされている任意のホストで、次のテンプレートを使用して pxGrid クライアント証明書署名要求を生成できるようになりました。

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
C = YOUR_COUNTRY
ST = YOUR_STATE
L = YOUR_CITY
O = YOUR_ORGANIZATION
OU = YOUR_ORGANIZATION_UNIT
CN = ise-connector.example.com
[v3_req]
subjectKeyIdentifier = hash
```

```
basicConstraints = critical,CA:false
subjectAltName = @alt_names
keyUsage = critical,digitalSignature,keyEncipherment
extendedKeyUsage = serverAuth,clientAuth
[alt_names]
IP.1 = 10.x.x.x
DNS.1 = ise-connector.example.com
```

ファイルを「example-connector.cfg」として保存し、ホストからOpenSSLコマンドを使用して、次のコマンドで証明書署名要求（CSR）と証明書の秘密キーを生成します。

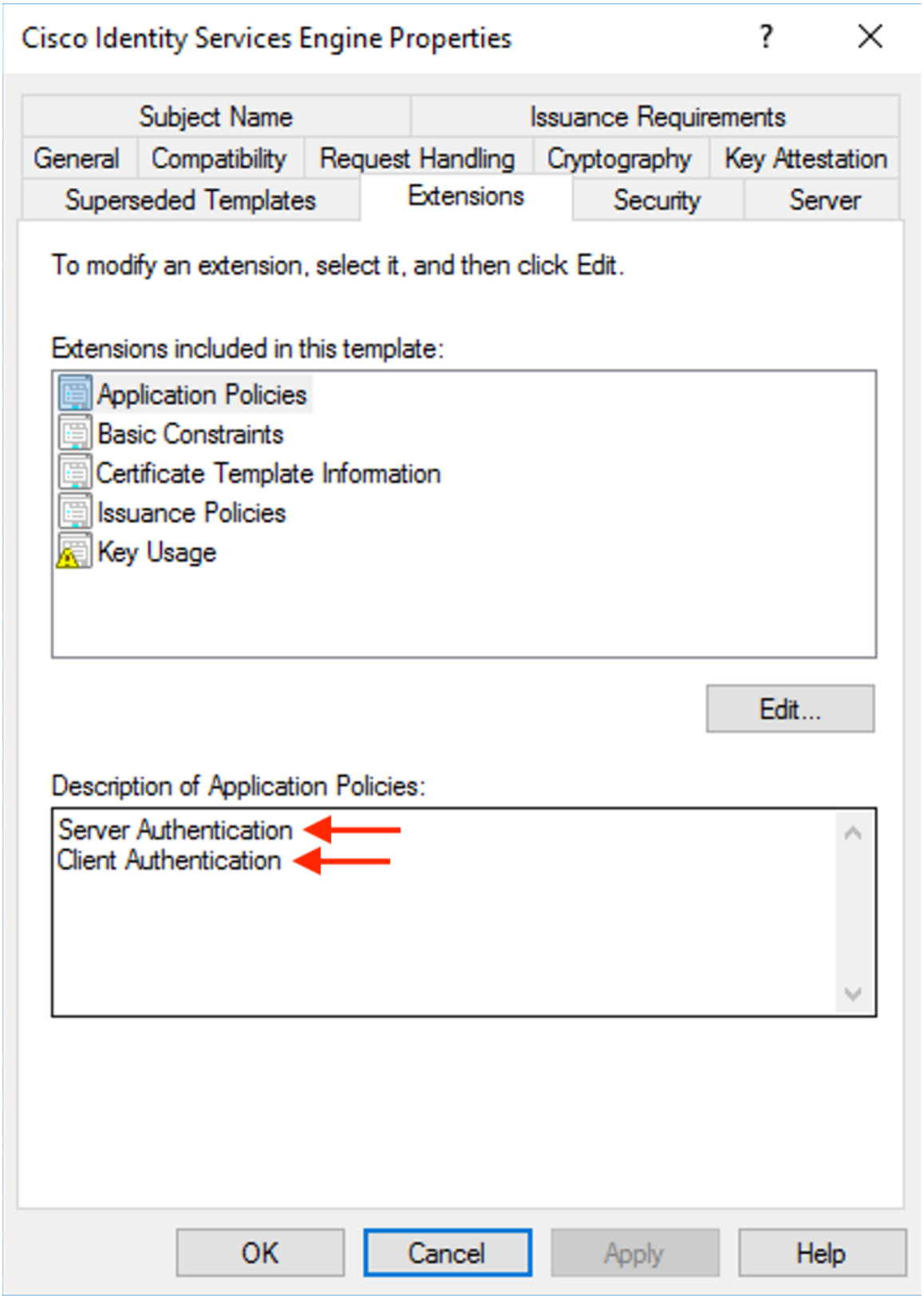
```
openssl req -newkey rsa:2048 -keyout example-connector.key -nodes -out
example-connector.csr -config example-connector.cfg
```

ステップ6 Windows CAサーバーを使用して、CAによって証明書署名要求（CSR）に署名します。Windows CA サーバーも使用している場合は、次のコマンドを実行して pxGrid クライアントの CSR に署名します。

```
certreq -submit -binary -attrib "CertificateTemplate:CiscoIdentityServicesEngine"
example-connector.csr example-connector.cer
```

（注） Windows CA には証明書テンプレートが必要です。このテンプレートには、次の拡張子が含まれている必要があります。

図 *68* : 証明書テンプレートのアプリケーションポリシーの拡張子



ステップ 7 署名されたクライアント証明書とルート CA を PEM 形式でホストにコピーします。これは、クライアント CSR と秘密キーを生成するホストと同じです。OpenSSL を使用して、クライアント証明書が X.509 PEM 形式であることを確認します。OpenSSL を使用して次のコマンドを実行し、署名されたクライアント証明書を X.509 PEM 形式に変換します。

```
openssl x509 -inform der -in example-connector.cer -out example-connector.pem
```

ステップ 8 次のコマンドを使用して、CA によって署名された PEM を確認することもできます。

```
openssl verify -CAfile root-ca.example.com.pem example-connector.pem
example-connector.pem: OK
```

（注） pxGrid を使用したマルチノード ISE 展開の場合、すべての pxGrid ノードは、Cisco Secure Workload ISE コネクタに使用される証明書を信頼する必要があります。

ステップ 9 上述の例のファイル名を使用して、ISE クライアント証明書（example-connector.pem）、クライアントキー（example-connector.key）および CA（root-ca.example.com.pem）を、以下に示すように、Cisco Secure Workload の ISE 設定ページのそれぞれのフィールドにコピーします。

（注） Cisco Secure Workload の最新バージョンにアップグレードする前に、ISE コネクタを削除して、既存の設定データが削除されていることを確認してください。アップグレードが完了したら、適用する新しいフィルタを使用して ISE コネクタを設定します。

図 *69 : ISE* コネクタ設定

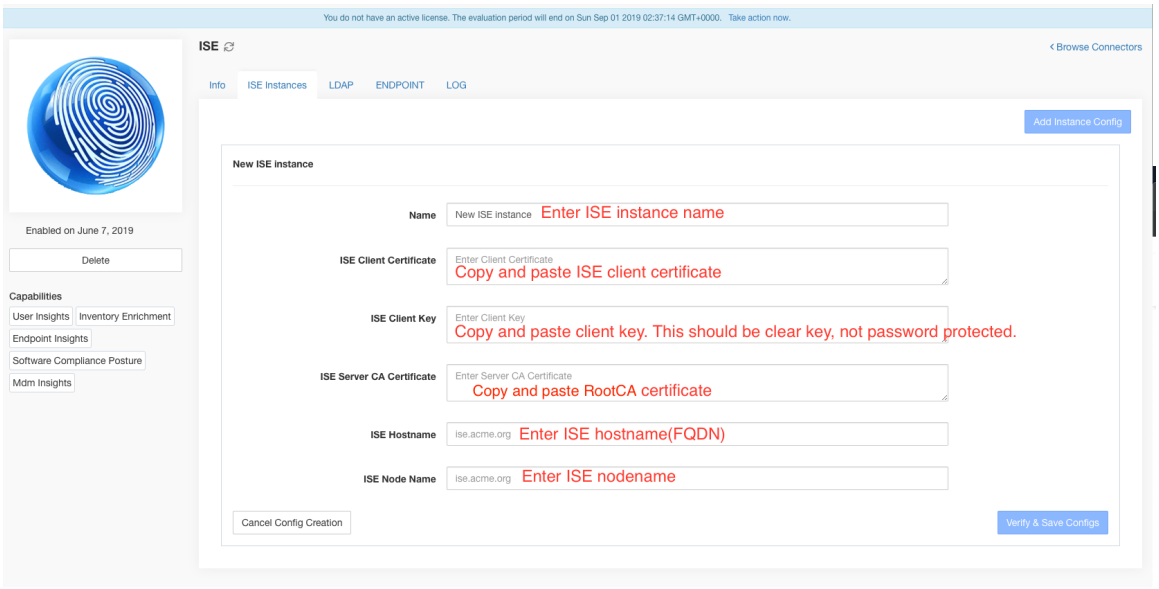


図 *70 : ISE* コネクタ設定

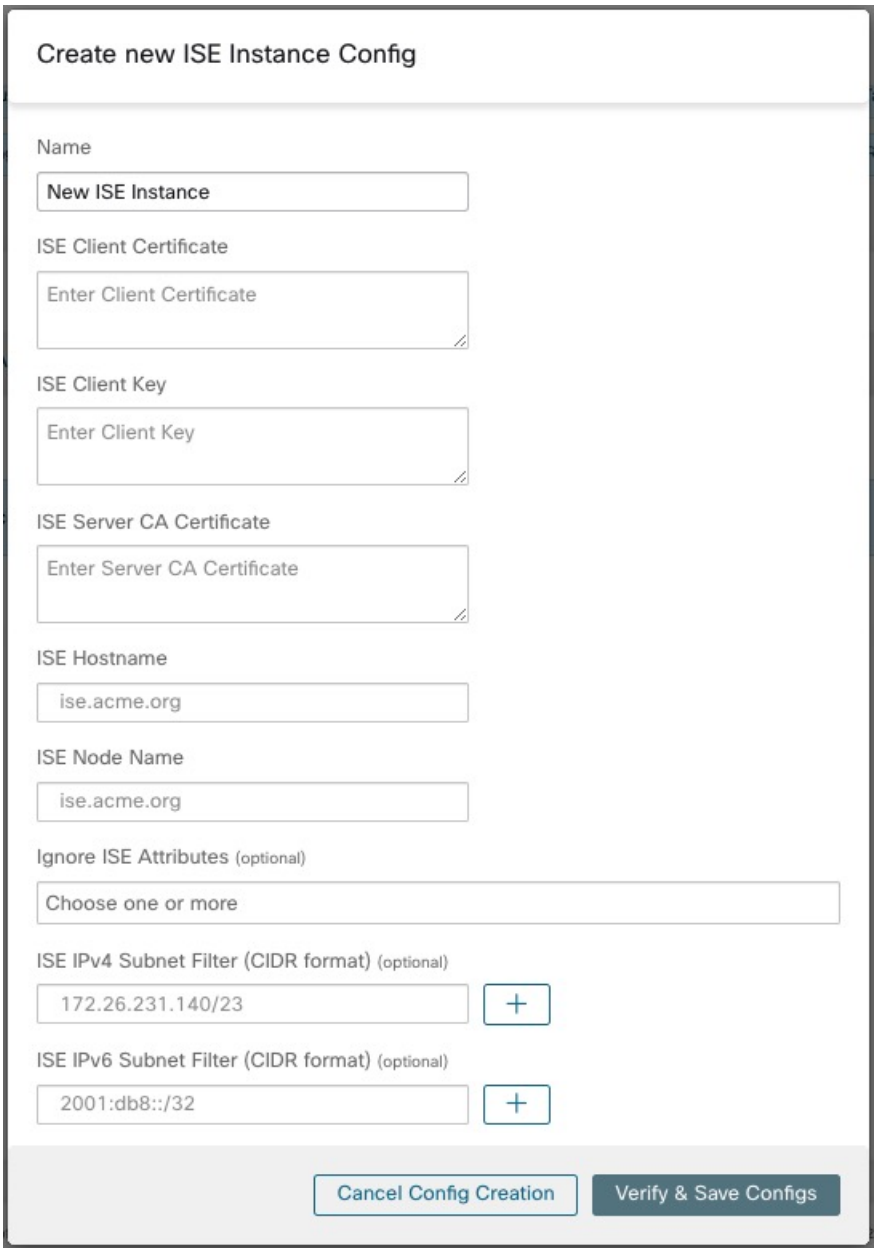


表 *20 : ISE* コネクタ設定

フィールド	説明
名前	ISE インスタンス名を入力します。
ISEクライアント証明書（ISE Client Certificate）	ISEクライアント証明書をコピーして貼り付けます。

フィールド	説明
ISEクライアントキー（ISE Client Key）	ISEクライアントキーをコピーして貼り付けます。クライアントキーは、パスワードで保護されていないクリアキーである必要があります。
ISEサーバーCA証明書（ISE Server CA Certificate）	ルートCA証明書をコピーして貼り付けます。
ISE のホスト名	ISE のホスト名（FQDN）を入力します。
ISEのノード名（ISE Node Name）	ISE のノード名を入力します。
ISE属性の無視（オプション）（Ignore ISE Attributes (Optional)）	リストから1つ以上のISE属性を選択します。 ISEを介してレポートされたエンドポイントのすべてのコンテキスト情報を取り込まない場合は、このオプションを使用します。
ISE IPv4サブネットフィルタ（CIDR形式）（オプション）（ISE IPv4 Subnet Filter (CIDR Format) (Optional)）	複数の IPv4 サブネットを入力して、ISE エンドポイントをフィルタします。
ISE IPv6サブネットフィルタ（CIDR形式）（オプション）（ISE IPv6 Subnet Filter (CIDR Format) (Optional)）	複数の IPv6 サブネットを入力して、ISE エンドポイントをフィルタします。

（注）

- ISE ホスト名に FQDN の代わりに IP アドレスが使用されている場合は、ISE CA 証明書 SAN の IP アドレスを使用します。そうしないと、接続が失敗する可能性があります。
- ISE のアクティブなエンドポイントの数はスナップショットではなく、ISE の設定とメトリックを計算するための集計期間によって異なります。Cisco Secure Workload のエージェント数は常に、ISE からの最後のプルおよび pxgrid 更新に基づくスナップショットであり、通常は過去1日間のアクティブなデバイス数です（フルスナップショットのデフォルトの更新頻度は 1 日です）。これらの数字は表記方法が異なるため、2 つの数字が必ずしも一致しないことがあります。

ISE レコードの処理

ISE コネクタは、次に説明するようにレコードを処理します。

エンドポイントレコード

ISE コネクタは ISE インスタンスに接続し、pxGrid を介してエンドポイントの更新をサブスクライブします。ISE コネクタがエンドポイントレコードを受信すると、そのエンドポイントは Cisco Secure Workload 上の ISE エージェントとして登録されます。ISE コネクタは、NVM レコードに存在するエンドポイント固有の情報と ISE コネクタの証明書を使用して、エンドポイントを登録します。エンドポイントが登録されたら、次のように動作します。ISE コネクタは、エンドポイントオブジェクトを Cisco Secure Workload のユーザーラベルとして送信することにより、このオブジェクトをインベントリ強化に使用します。ISE コネクタが切断されたエンドポイントを ISE から取得すると、Cisco Secure Workload からインベントリ強化が削除されます。

セキュリティ グループ レコード

ISE コネクタは、pxGrid を介してセキュリティ グループ ラベルの変更に関する更新もサブスクライブします。このレコードを受信すると、ISE コネクタはローカルデータベースを維持します。このデータベースを使用して、エンドポイントレコードの受信時に SGT 名を値にマッピングします。

定期的タスク

ISE コネクタは、ISE エンドポイントインベントリでユーザーラベルを定期的に共有します。

1. [エンドポイントスナップショット（Endpoint Snapshots）] : 20 時間ごとに、ISE コネクタは、エンドポイントとセキュリティグループラベルのスナップショットを ISE インスタンスから取得し、変更が検出された場合はクラスタを更新します。このコールでは、ISE からの Secure Workload でエンドポイントが表示されない場合に切断されるエンドポイントは検出対象になりません。
2. [ユーザーラベル（User Labels）] : 2 分ごとに、ISE コネクタは、ローカルで保持されている LDAP ユーザーと ISE エンドポイントラベルをスキャンし、それらの IP アドレスのユーザーラベルを更新します。

ユーザーラベル用に、ISE コネクタは、組織に属する全ユーザーの LDAP 属性のローカルスナップショットを作成します。ISE コネクタが有効になっている場合、LDAP の設定（サーバー/ポート情報、ユーザーに関して取得される属性、ユーザー名を含む属性）が提供される場合があります。さらに、LDAP サーバーにアクセスするための LDAP ユーザーログイン情報が提供されることもあります。LDAP ユーザーログイン情報は暗号化され、ISE コネクタで公開されることはありません。オプションで、LDAP サーバーに安全にアクセスするための LDAP 証明書の提供も可能です。

（注） ISE コネクタは、24 時間ごとに新しいローカル LDAP スナップショットを作成します。この間隔は、コネクタの LDAP 設定で変更できます。

（注） Cisco ISE デバイスのアップグレードでは、アップグレード後に ISE によって生成された新しい証明書を使用して ISE コネクタを再設定する必要があります。

制限

メトリック	制限
1 つの ISE コネクタで設定可能な ISE インスタンスの最大数	20
1 つの Secure Workload Edge アプライアンスにおける ISE コネクタの最大数	1
1 つのテナント（rootscope）における ISE コネクタの最大数	1
Cisco Secure Workload における ISE コネクタの最大数	150

インベントリ強化用のコネクタ

インベントリ強化用のコネクタは、Secure Workload によって監視されるインベントリ（IP アドレス）に関する追加のメタデータとコンテキストを提供します。

コネクタ	説明	仮想アプライアンス上に展開
ServiceNow	ServiceNow インスタンスからエンドポイント情報を収集し、ServiceNow 属性でインベントリを強化します。	Secure Workload Edge
関連項目：	Cloud Connector	–

必要な仮想アプライアンスの詳細については、「コネクタ用の仮想アプライアンス」を参照してください。

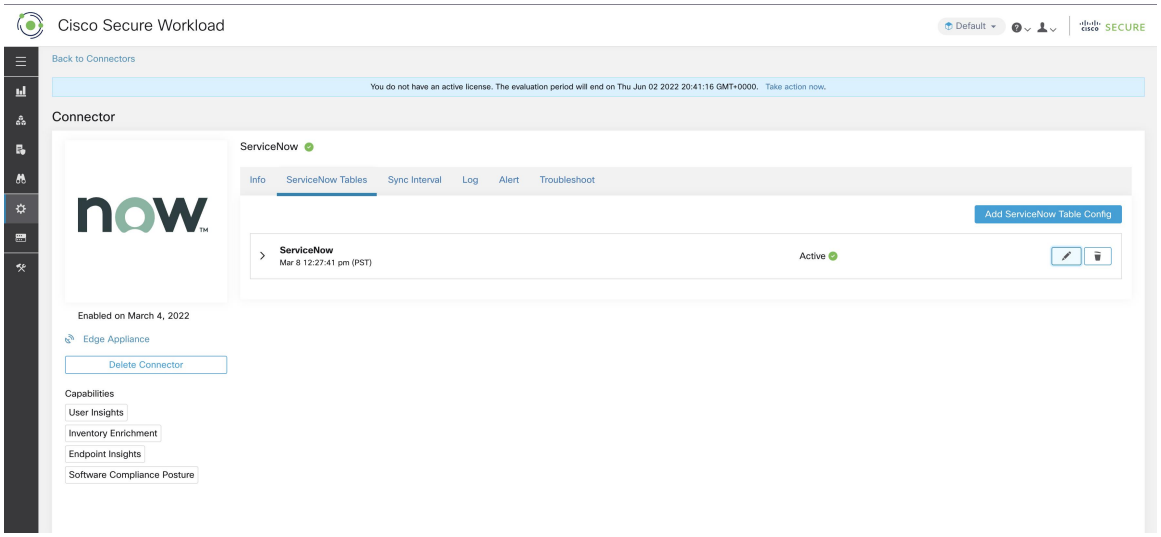
ServiceNow コネクタ

ServiceNow コネクタは ServiceNow インスタンスに接続して、ServiceNow インベントリ内にあるエンドポイントのすべての ServiceNow CMDB 関連ラベルを取得します。このソリューションを使用すると、Cisco Secure Workload のエンドポイントに関する豊富なメタデータを取得できます。

ServiceNow コネクタは、次の高度な機能を実行します。

1. これらのエンドポイントの Cisco Secure Workload のインベントリで ServiceNow メタデータを更新します。
2. 定期的にスナップショットを取り、これらのエンドポイントのラベルを更新します。

図 *71 : ServiceNow* コネクタ



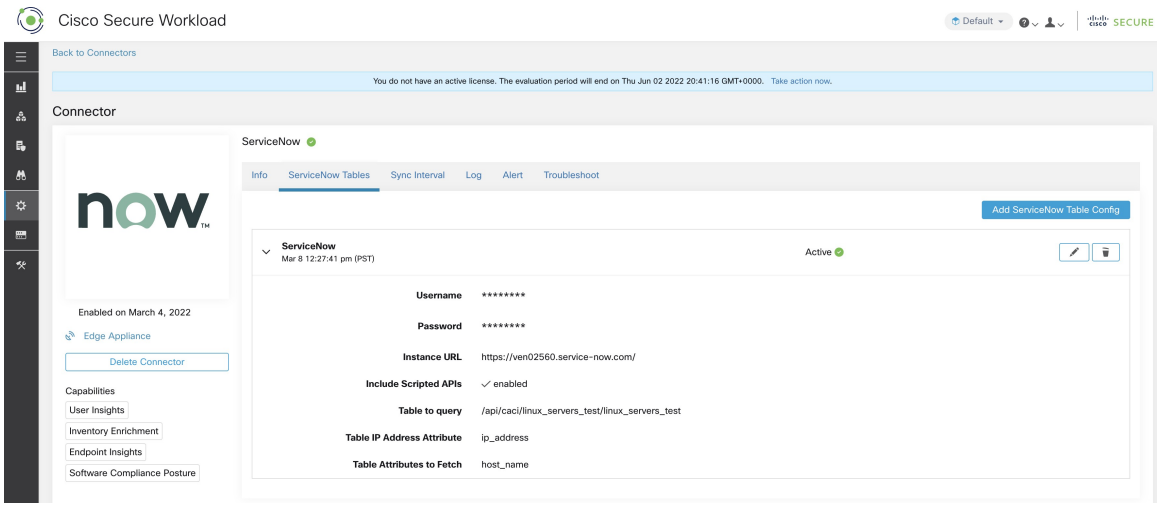
ServiceNow コネクタの設定方法

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。コネクタでは、以下を設定できます。

- *ServiceNow* テーブル：ServiceNow インスタンスのクレデンシャルと、データを取得する ServiceNow テーブルに関する情報を設定します。
- スクリプト化された *REST API*：ServiceNow のスクリプト化された REST API テーブルは、ServiceNow テーブルと同様に設定できます。
- 同期間隔：Secure Workload がServiceNow インスタンスに対して更新されたデータについてのクエリを実行する周期を変更できます。
- ログ：詳細については、「ログ設定」を参照してください。

ServiceNow インスタンスの構成

図 *72 : ServiceNow* インスタンスの構成



ServiceNow インスタンスを正常に構成するには、次の項目が必要です。

1. ServiceNow ユーザー名
2. ServiceNow パスワード
3. ServiceNow インスタンスの URL
4. スクリプト化された API を含める

必要な項目が準備できたら、Secure Workload は ServiceNow インスタンスおよび Scripted REST API から（[スクリプト化されたAPIを含める（Include Scripted APIs）]チェックボックスが有効になっている場合のみ）すべてのテーブルの検出を実行します。選択するテーブルのリストがユーザーに提示され、ユーザーがテーブルを選択すると、Secure Workload はそのテーブルから属性のすべてのリストを取得して、ユーザーが選択できるようにします。ユーザーは、キーとして ip_address 属性をテーブルから選択する必要があります。その後、ユーザーはテーブルから最大 10 個の一意の属性を選択できます。各ステップについては、次の図を参照してください。

（注） ServiceNow コネクタは、[IPアドレス（IP Address）] フィールドを持つテーブルとの統合のみをサポートできます。

（注） ServiceNow Scripted REST API と統合するには、[スクリプト化されたAPI（Scripted APIs）] チェックボックスを有効にする必要があります。これにより、他のテーブルと同様のワークフローが得られます。

（注） Scripted REST API を ServiceNow コネクタと統合する場合、パスパラメータを含めることはできません。また、APIはクエリパラメータとして**sysparm_limit**、**sysparm_fields**、**sysparm_offset**をサポートする必要があります。

（注） ServiceNow ユーザーロールには、テーブル用の **cmdb_read** と、 Scripted REST API を Cisco Secure Workload と統合するための **web_service_admin** を含める必要があります。

図 *73 : ServiceNow* インスタンス構成の最初のステップ

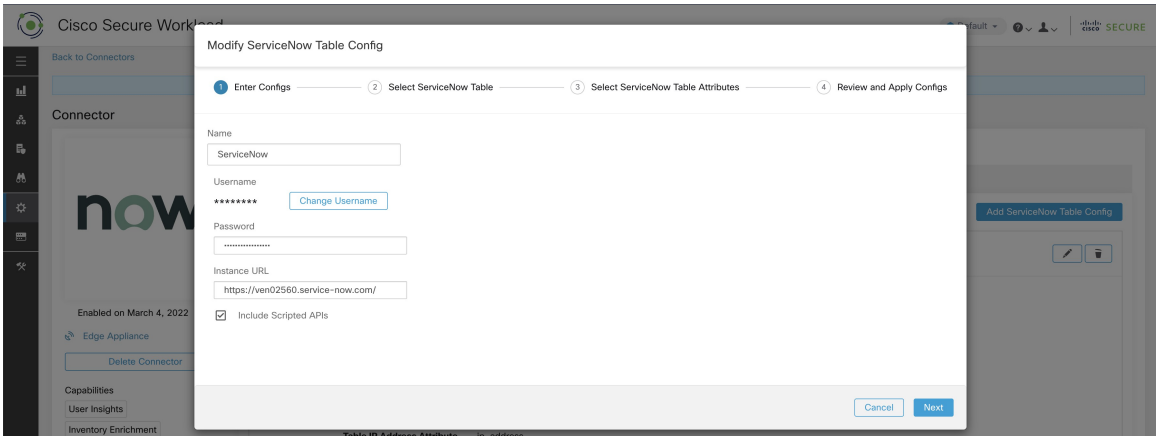


図 *74 : Secure Workload* が *ServiceNow* インスタンスからテーブル情報を取得します

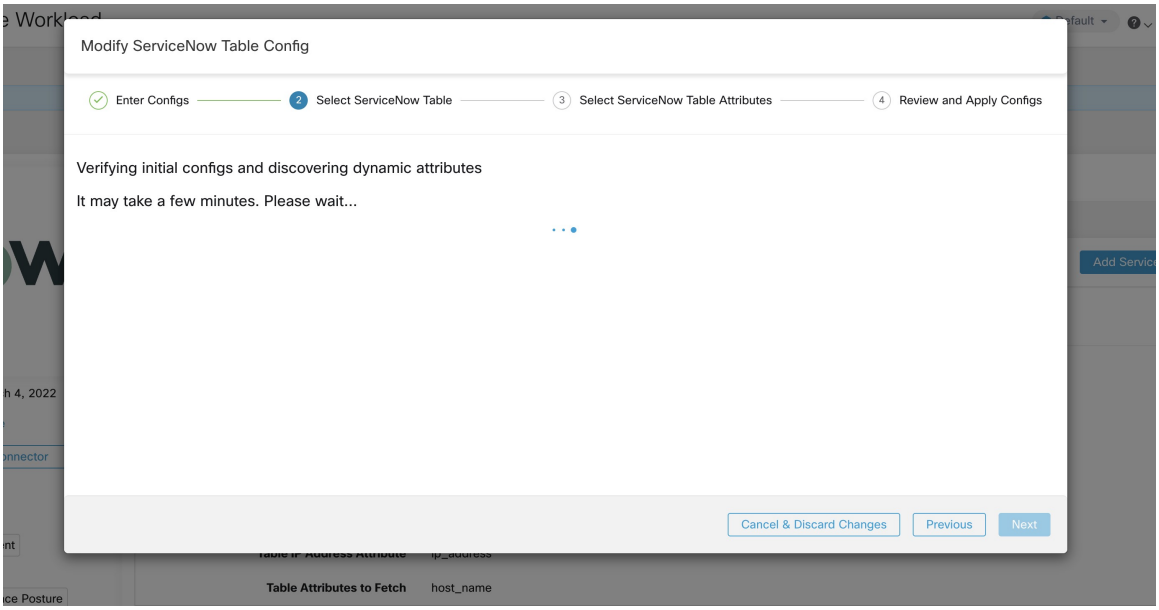


図 *75 : Secure Workload* がテーブルのリストを提示します

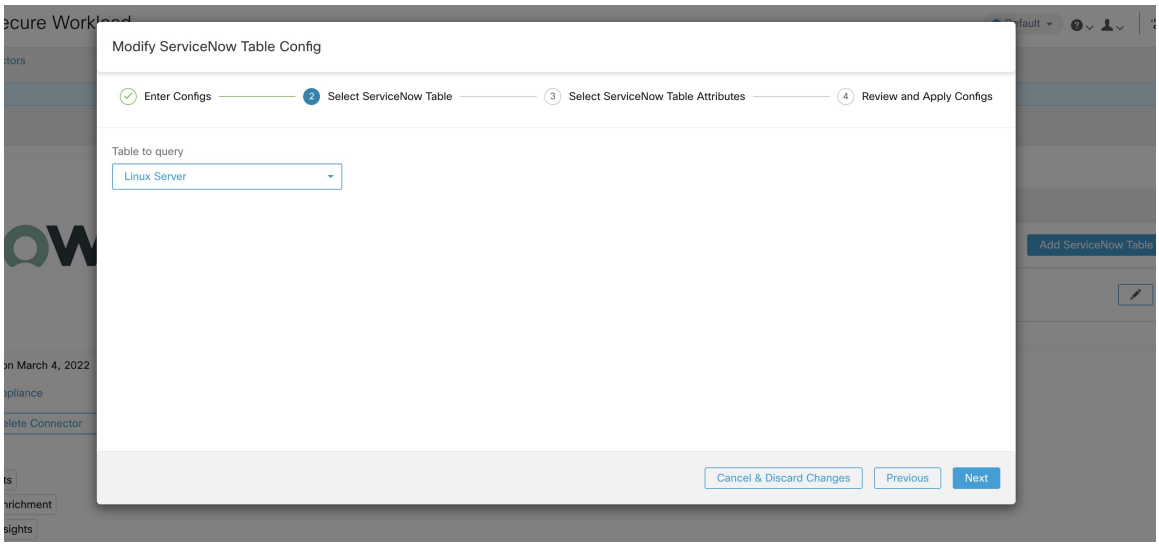


図 *76 :* ユーザーが *ip_address* 属性とテーブル内の他の属性を選択します

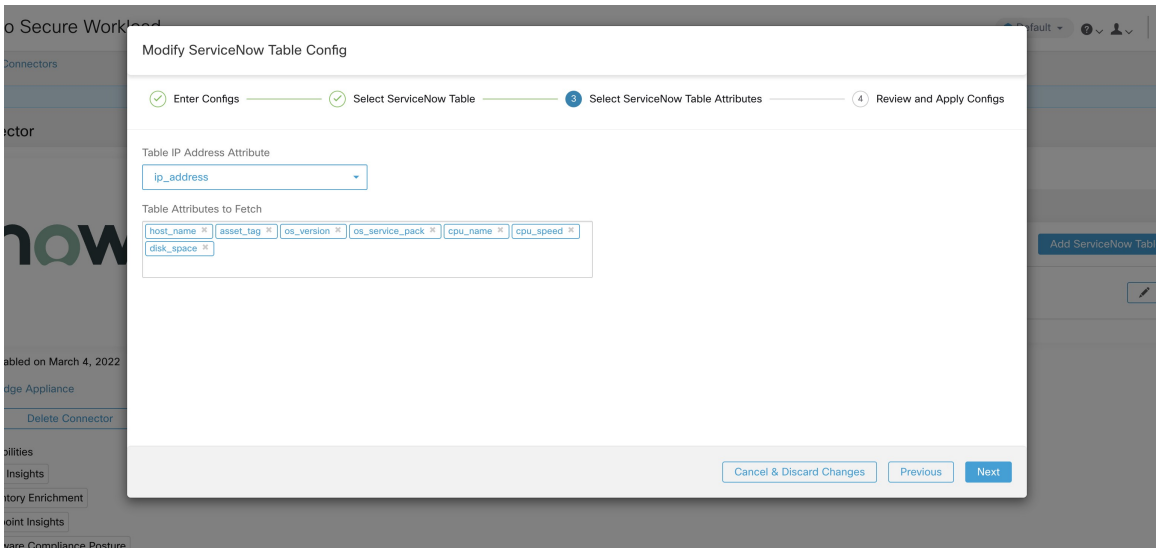
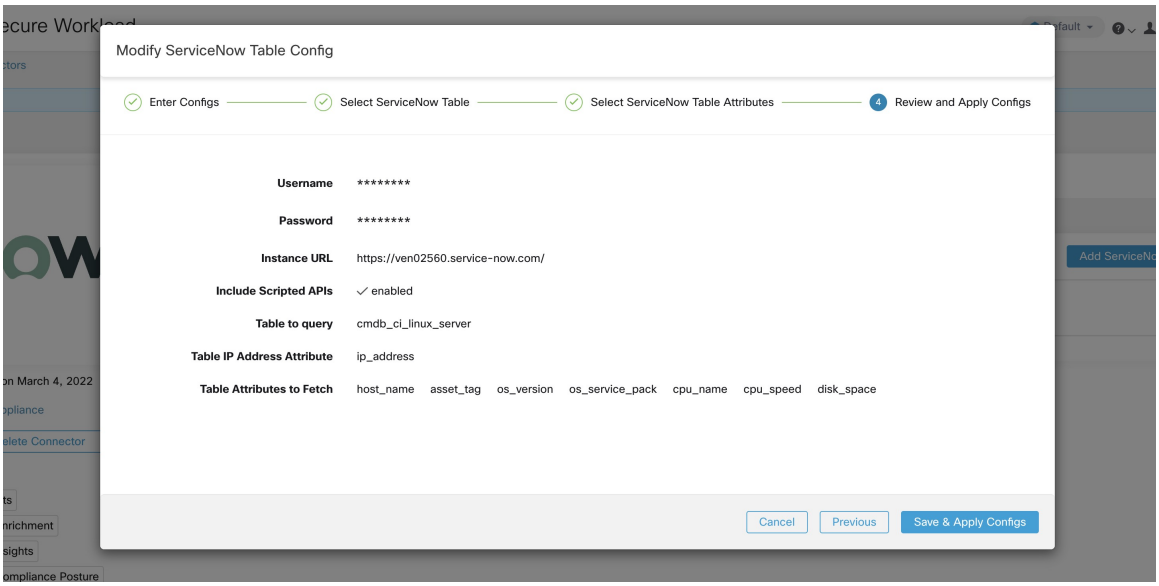


図 *77* : ユーザーが ***ServiceNow*** 構成を完了します



ServiceNow レコードの処理

ServiceNow コネクタは ServiceNow インスタンスに接続し、設定されたテーブルに基づいて、それらのテーブルに対してクエリを実行して ServiceNow ラベル/メタデータを取得します。Secure Workload は ServiceNow ラベルにそのインベントリ内の IP アドレスの注釈を付けます。ServiceNow コネクタは定期的に新しいラベルを取得し、Secure Workload インベントリを更新します。

（注） Secure Workload は ServiceNow テーブルから定期的にレコードを取得します。これは、ServiceNow コネクタの [SyncInterval] タブで構成できます。デフォルトの同期間隔は 60 分です。エントリ数が多い ServiceNow テーブルと統合する場合は、この同期間隔をより高い値に設定する必要があります。

（注） Secure Workload は、10 回の連続的な同期間隔で確認されなかったエントリを削除します。ServiceNow インスタンスへの接続が長時間停止した場合、そのインスタンスのすべてのラベルがクリーンアップされる可能性があります。

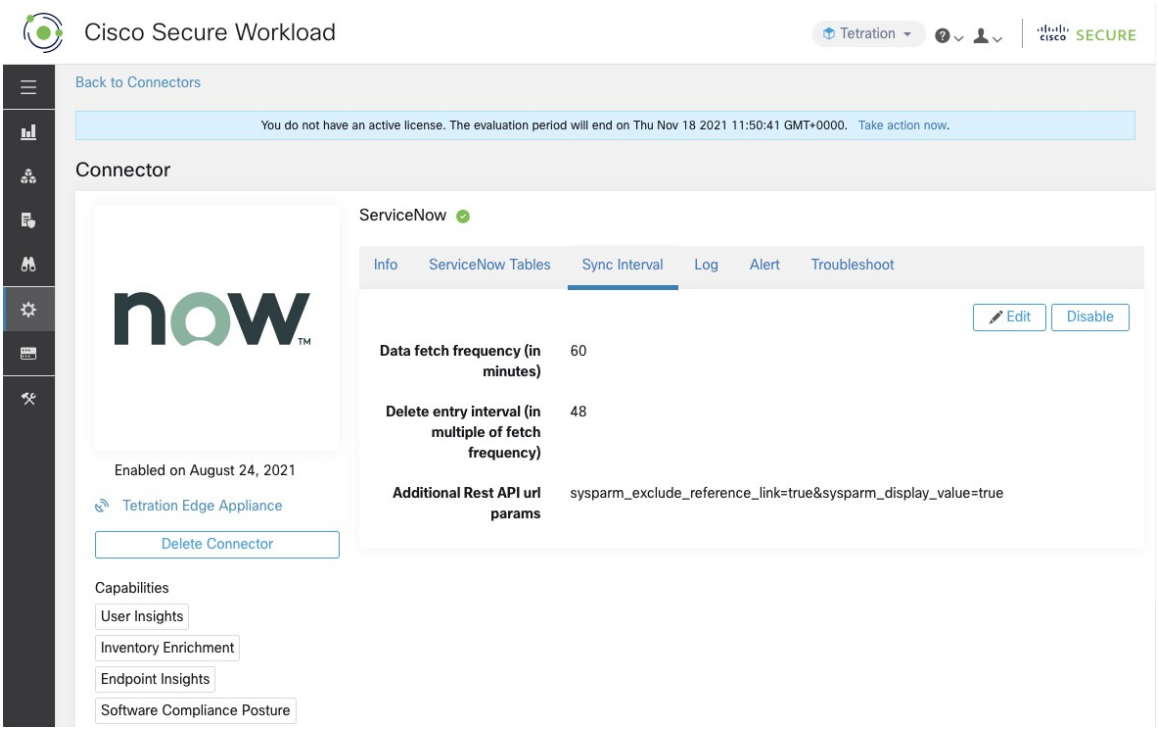
同期間隔の設定

1. Cisco Secure Workload ServiceNow コネクタでは、 Secure Workload と ServiceNow インスタンス間の同期の頻度を設定できます。デフォルトでは、同期間隔は 60 分に設定されていますが、同期間隔の設定で [データ取得頻度（Data fetch frequency）] として変更できます。

2. レコードの削除の検出について、Secure Workload ServiceNow コネクタは ServiceNow インスタンスとの同期に依存しています。48 回の連続した同期間隔でエントリが見つからない場合は、先に進んでエントリを削除します。エントリの削除間隔は、同期間隔設定で [エントリの削除間隔（Delete entry interval）] として設定できます。

3. ServiceNow テーブルの REST API を呼び出すときに追加のパラメータが渡される場合は、*Additional Rest API url params* の一部としてそれらを設定できます。この設定は任意です。たとえば、URL パラメータ **sysparm_exclude_reference_link=true&sysparm_display_value=true** を使用して、ServiceNow から参照ルックアップを取得できます。

図 *78* : 同期間隔の設定



ラベルを削除する Explore コマンド

ユーザーが特定のインスタンスの特定の IP のラベルを（削除間隔を待たずに）すぐにクリーンアップする場合は、explore コマンドを使用して実行できます。コマンドを実行する手順は次のとおりです。

1. テナントの VRF ID の検索
2. Explore コマンド UI へのアクセス
3. コマンドの実行

テナントの VRF ID の検索

サイト管理者および**カスタマーサポート**ユーザーは、ウィンドウの左側にあるナビゲーションバーの [プラットフォーム（Platform）] メニューの下にある [テナント（Tenant）] ページにアクセスできます。このページには、現在構成されているすべてのテナントと VRF が表示されます。詳細については、「テナント」の項を参照してください。

[テナント（Tenants）] ページの [テナント（Tenants）] テーブルの ID フィールドは、テナントの VRF ID です。

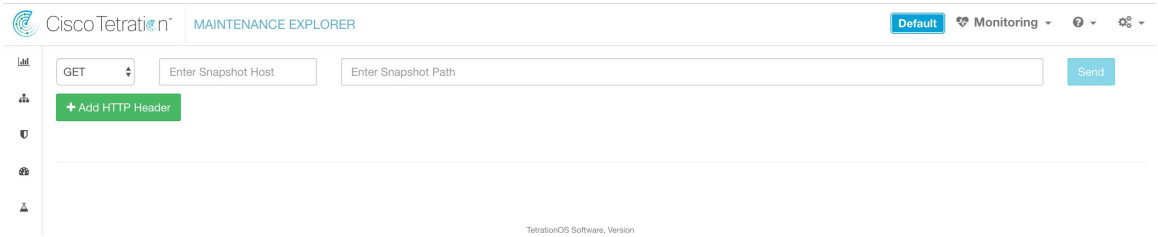
Explore コマンド UI へのアクセス

[メンテナンスエクスプローラ（Maintenance Explorer）] コマンドインターフェイスにアクセスするには、Secure Workload Web インターフェイスの左側のナビゲーションバーから **[トラブルシュート（Troubleshoot）] > [メンテナンスエクスプローラ（Maintenance Explorer）]** を選択します。

（注） エクスプローラメニューにアクセスするには、カスタマーサポートの権限が必要です。エクスプローラタブが表示されない場合は、アカウントに必要な権限がない可能性があります。

ドロップダウンメニューのエクスプローラタブをクリックして、[メンテナンスエクスプローラ（Maintenance Explorer）] ページに移動します。

図 *79 : [メンテナンスエクスプローラ（Maintenance Explorer）] タブ*



コマンドの実行

- アクションとして `POST` を選択します。
- スナップショットホストとして「`orchestrator.service.consul`」と入力します。
- スナップショットパスを入力します。

  servicenow インスタンスの特定の IP のラベルを削除するには：
  `servicenow_cleanup_annotations?args=<vrf-id><ip_address><instance_url><table_name>`
- [送信] をクリック

（注） explore コマンドを使用して削除した後、ServiceNow インスタンスにレコードが表示された場合は、再入力されます。

FAQ

1. **ServiceNow CMDB テーブルに IP アドレスがない場合はどうなりますか。**

   IP アドレスがない場合、現在のテーブルの必要なフィールドと IP アドレス（別のテーブルとの JOIN 操作から取得される可能性がある）を持つ View on ServiceNow を作成することを推奨します。このようなビューが作成されると、テーブル名の代わりに使用できます。

2. **ServiceNow インスタンスに MFA が必要な場合はどうなりますか。**

   現在、MFA を使用した ServiceNow インスタンスとの統合はサポートしていません。

制限

メトリック	制限
1 つの ServiceNow コネクタで構成できる ServiceNow インスタンスの最大数	20
1 つの ServiceNow インスタンスから取得できる属性の最大数	10
1 つの Secure Workload Edge アプライアンス上の ServiceNow コネクタの最大数	1
1 つのテナント（ルート範囲）上の ServiceNow コネクタの最大数	1
Cisco Secure Workload 上の ServiceNow コネクタの最大数	150

アラート通知用のコネクタ

アラート通知用のコネクタを使用すると、Secure Workload はさまざまなメッセージングおよびロギングプラットフォームで Secure Workload アラートを発行できます。このコネクタは、Secure Workload Edge アプライアンスの TAN サービスで稼働します。

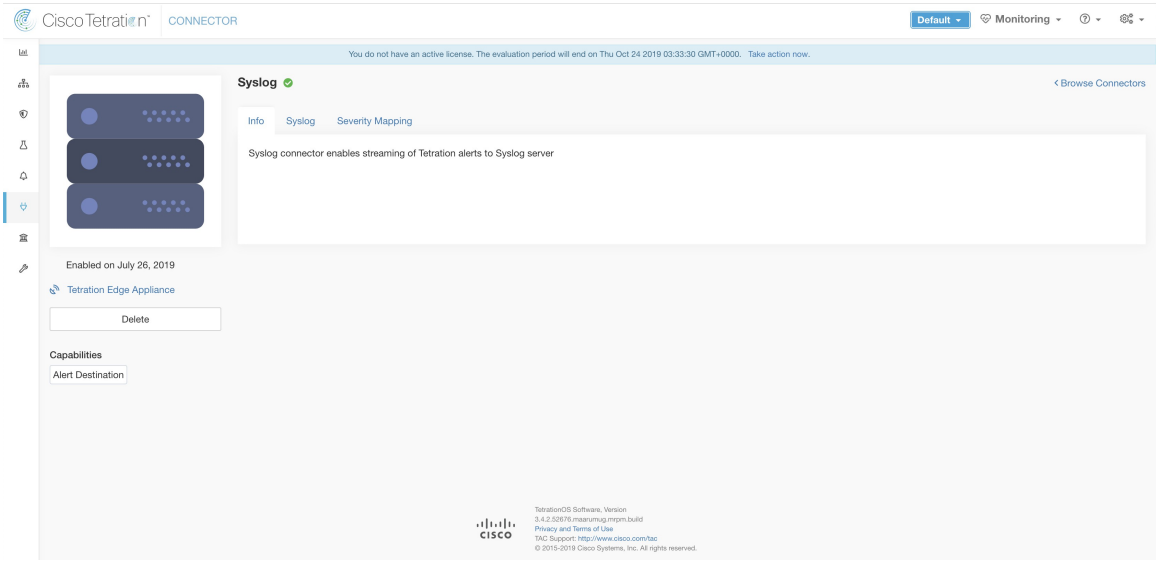
コネクタ	説明	仮想アプライアンス上に展開
Syslog	Syslog サーバーに Secure Workload アラートを送信します。	Secure Workload Edge
E メール	電子メールで Secure Workload アラートを送信します。	Secure Workload Edge
Slack	Slack で Secure Workload アラートを送信します。	Secure Workload Edge
Pager Duty	Pager Duty で Secure Workload アラートを送信します。	Secure Workload Edge
Kinesis	Amazon Kinesis で Secure Workload アラートを送信します。	Secure Workload Edge

必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。

Syslog Connector

有効にすると、Cisco Secure Workload Edge アプライアンスの TAN サービスは、構成を使用してアラートを Syslog サーバーに送信できます。

図 *80 : Syslog Connector*



次の表は、Syslog サーバーで Secure Workload アラートを公開するための構成の詳細を示しています。詳細については、「Syslog 通知設定」を参照してください。

パラメータ名	タイプ	説明
Protocol	dropdown • [UDP] • [TCP]	サーバーへの接続に使用するプロトコル。
Server Address	string	Syslog サーバーの IP アドレスまたはホスト名。
ポート（**Port**）	number	Syslog サーバーのリスニングポート。デフォルトのポート値は 514 です。

図 ***81 : Syslog Connector*** の設定例

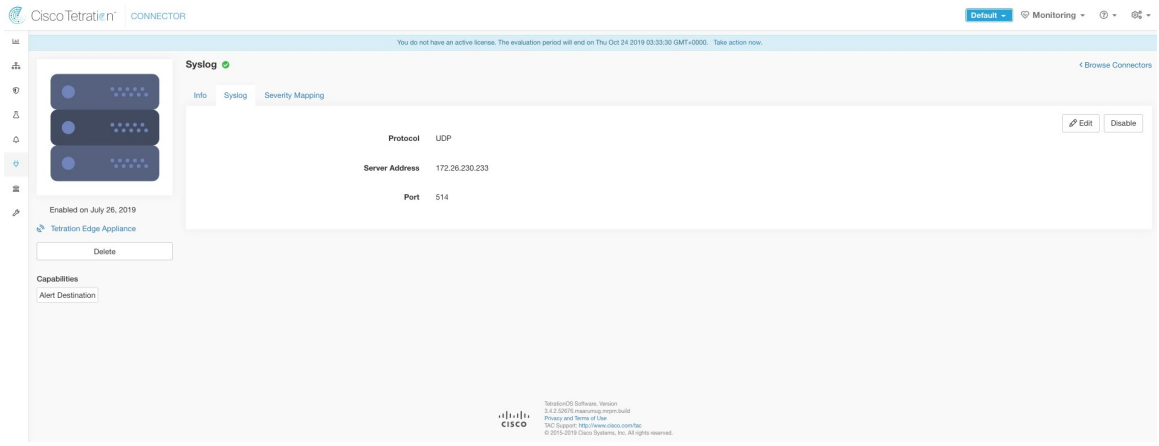


図 ***82 :*** サンプルアラート



Syslog のシビラティ（重大度）のマッピング

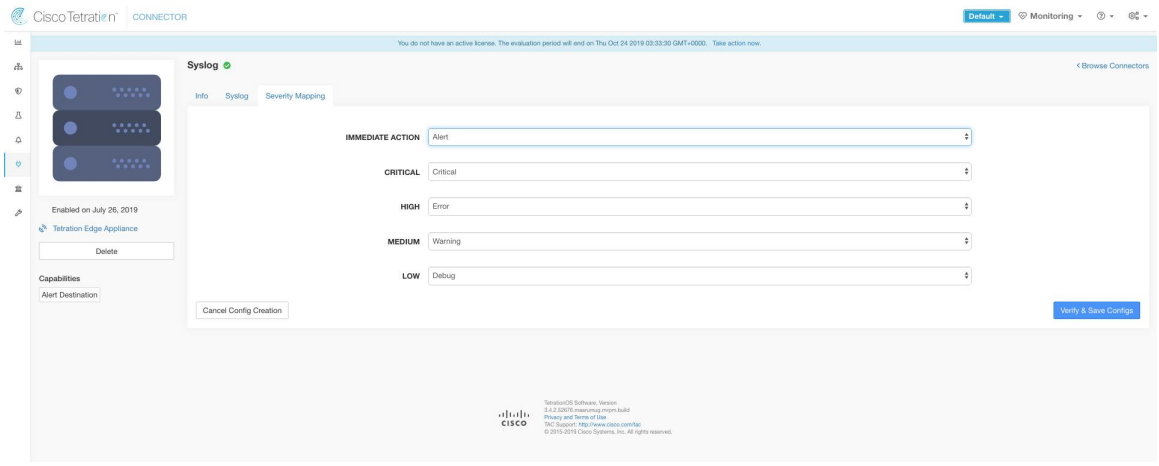
次の表は、Syslog の Secure Workload アラートにおけるデフォルトのシビラティ（重大度）マッピングを示しています。

安全なワークロードアラートのシビラティ（重大度）	Syslog のシビラティ（重大度）
LOW	LOG_DEBUG
[中（Medium）]	LOG_WARNING
HIGH	LOG_ERR
CRITICAL	LOG_CRIT
即時対応（IMMEDIATE ACTION）	LOG_EMERG

この設定は、Syslog コネクタの[シビラティ（重大度）マッピング（Severity Mapping）]設定を使用して変更できます。Secure Workload アラートのシビラティ（重大度）ごとに対応する Syslog 優先度を選択し、シビラティ（重大度）マッピングを変更できます。詳細については、「Syslog のシビラティ（重大度）のマッピング設定」を参照してください。

パラメータ名	マッピングのドロップダウン
[即時対応（IMMEDIATE_ACTION）]	• *[緊急（Emergency）]* • *[アラート（Alert）]* • *[Critical]* • *[エラー（Error）]* • [警告（Warning）] • *[通知（Notice）]* • *[Informational]* • デバッグ（*Debug*）
CRITICAL	
HIGH	
[中（Medium）]	
LOW	

図 *83 : Syslog* のシビラティ（重大度）のマッピング例



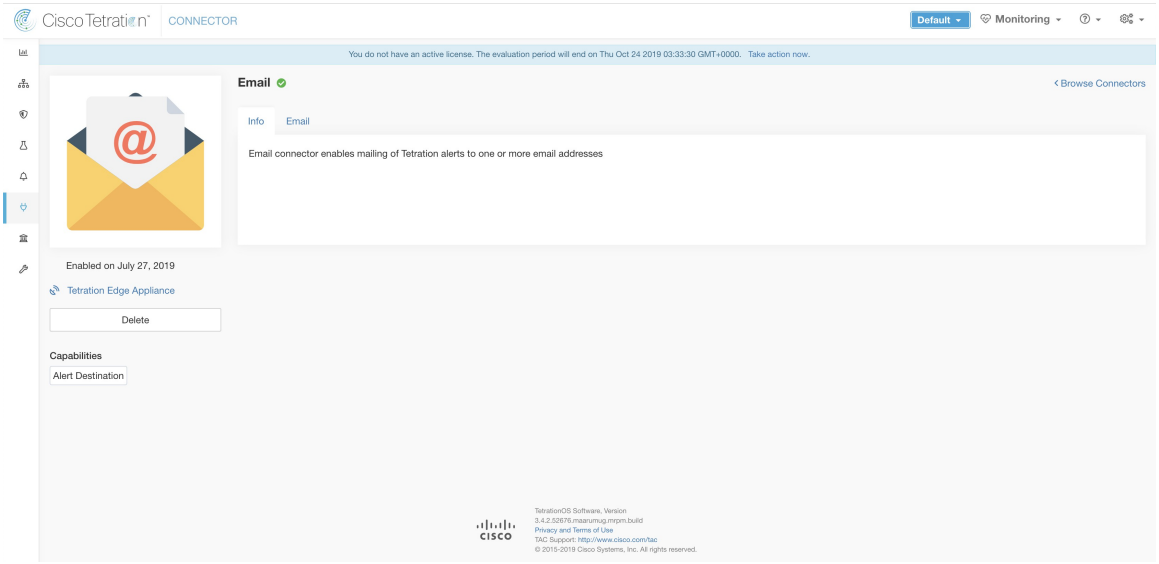
制限

メトリック	制限
1 つの Secure Workload Edge アプライアンスにおける Syslog コネクタの最大数	1
1 つのテナント（ルート範囲）における Syslog コネクタの最大数	1
Cisco Secure Workload における Syslog コネクタの最大数	150

Email Connector

有効にすると、Secure Workload Edge アプライアンスの TAN サービスは、指定された構成にアラートを送信できます。

図 *84 : Email Connector*



次の表は、電子メールで Secure Workload アラートを公開するための構成の詳細を示しています。詳細については、「電子メール通知設定」を参照してください。

表 *21 :* 電子メール通知設定の詳細

パラメータ名	タイプ	説明
SMTP ユーザ名（SMTP Username）	string	SMTP サーバーのユーザー名。このパラメータはオプションです。
SMTP パスワード（SMTP Password）	string	ユーザーの SMTP サーバーパスワード（指定されている場合）。このパラメータはオプションです。
SMTP Server	string	SMTP サーバーの IP アドレスまたはホスト名
SMTP Port	number	SMTP サーバーのリスニングポート。デフォルト値は 587 です。
[Secure Connection]	チェックボックス	SMTP サーバー接続に SSL を使用する必要があるかどうか。
From Email Address	string	アラートの送信に使用する電子メールアドレス

パラメータ名	タイプ	説明
[デフォルト受信者（Default Recipients）]	string	カンマで区切られた受信者の電子メールアドレス一覧。

図 *85 : Email Connector* の設定例

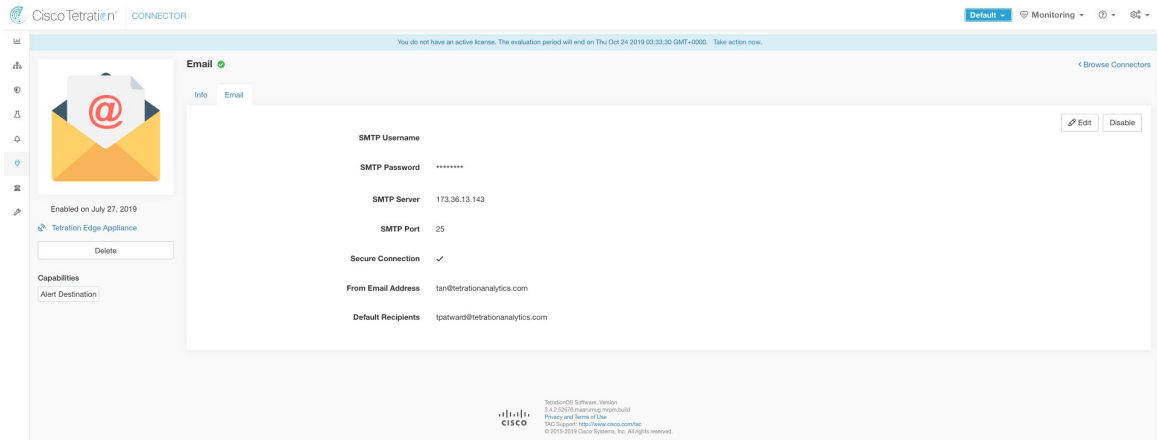
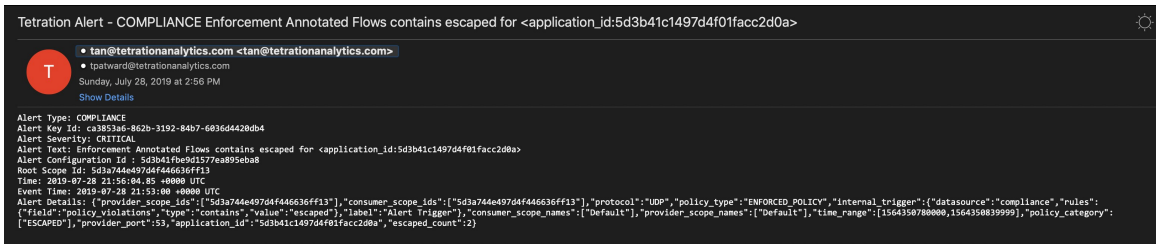


図 *86 :* サンプルアラート



（注）

- SMTP ユーザー名やパスワードの指定は任意です。ユーザー名が指定されていない場合、認証なしで SMTP サーバーに接続しようとします。
- [セキュアな接続（Secure Connection）] ボックスがオンになっていない場合、セキュアでない接続を介してアラート通知を送信します。
- デフォルトの受信者リストは、アラート通知の送信に使用されます。このリストは、アラート設定で必要な場合、アラートごとにオーバーライドできます。

制限

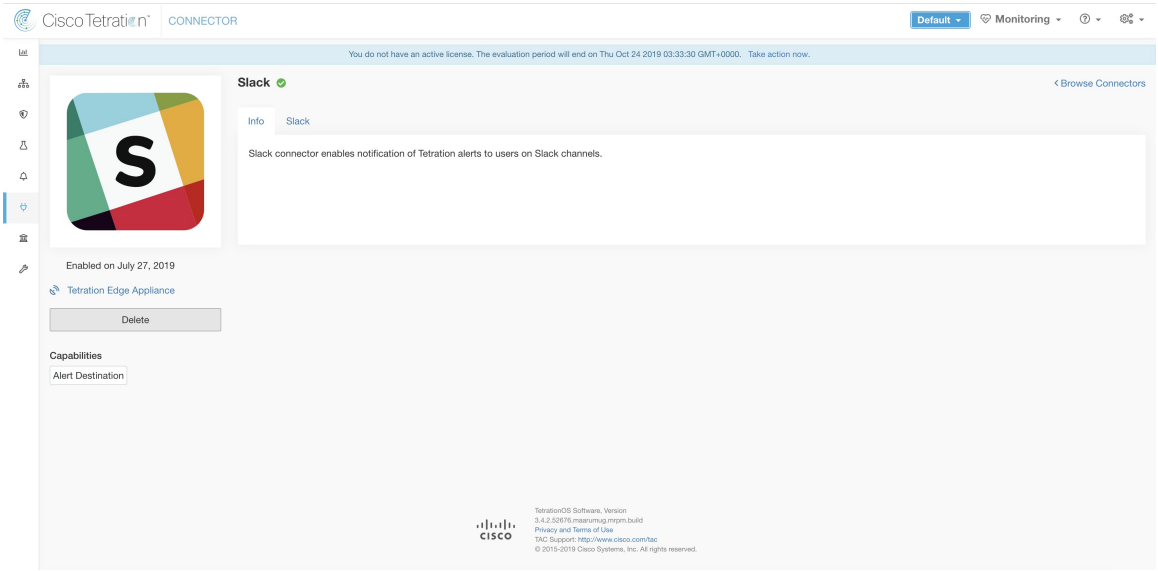
メトリック	制限
1 つの Secure Workload Edge アプライアンスにおける電子メールコネクタの最大数	1

メトリック	制限
1 つのテナント（ルート範囲）における電子メールコネクタの最大数	1
Cisco Secure Workload における電子メールコネクタの最大数	150

Slack コネクタ

有効にすると、Secure Workload Edge アプライアンスの TAN サービスは、設定に基づいてアラートを Slack に送信できます。

図 *87 : Slack* コネクタ



次の表は、 Slack で Secure Workload アラートを発行するための設定の詳細を示しています。詳細については、

「Slack 通知設定」を参照してください。

パラメータ名	タイプ	説明
Slack ウェブフック URL	string	Secure Workload アラートを発行する Slack ウェブフック

（注）
- Slack ウェブフックを生成する方法は、こちらを参照してください。

図 *88 : Slack* コネクタの設定例

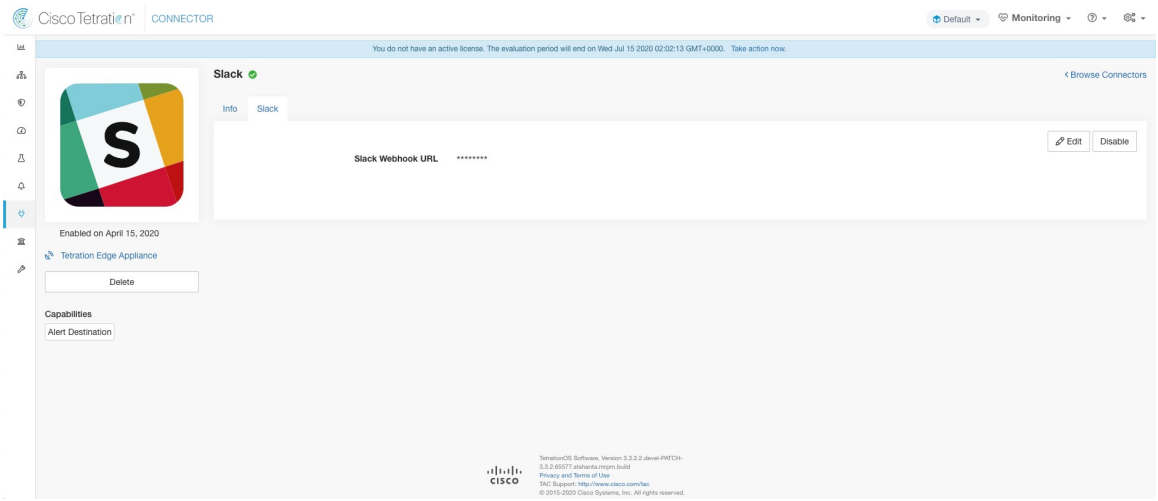
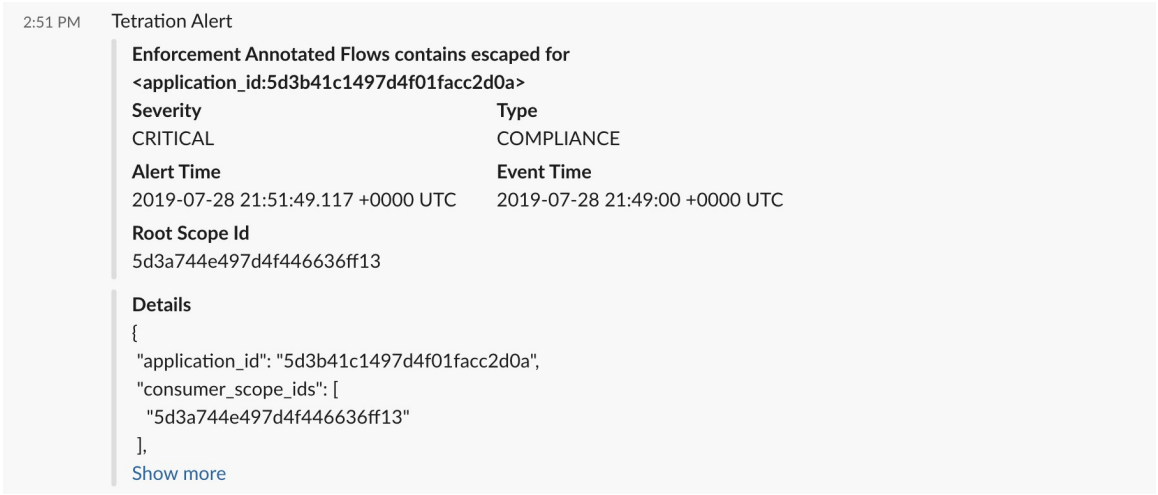


図 *89 :* サンプルアラート



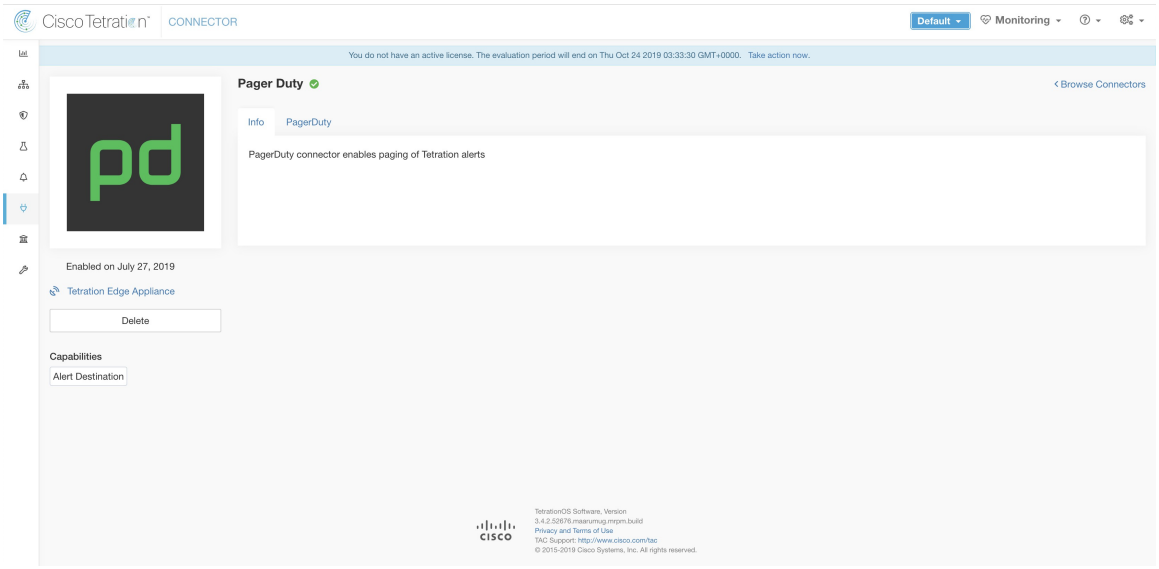
制限

メトリック	制限
1 つの Secure Workload Edge アプライアンスにおける Slack コネクタの最大数	1
1 つのテナント（ルート範囲）における Slack コネクタの最大数	1
Cisco Secure Workload における Slack コネクタの最大数	150

PagerDuty Connector

有効にすると、Secure Workload Edge アプライアンスの TAN サービスは、この構成を使用して PagerDuty にアラートを送信できます。

図 *90 : PagerDuty* コネクタ



次の表は、PagerDuty で Secure Workload アラートを発行するための構成の詳細を示しています。詳細については、「PagerDuty 通知設定」を参照してください。

パラメータ名	タイプ	説明
PagerDuty サービスキー（PagerDuty Service Key）	string	PagerDuty で Secure Workload アラートをプッシュするための PagerDuty サービスキー

図 *91 : PagerDuty* コネクタの構成例

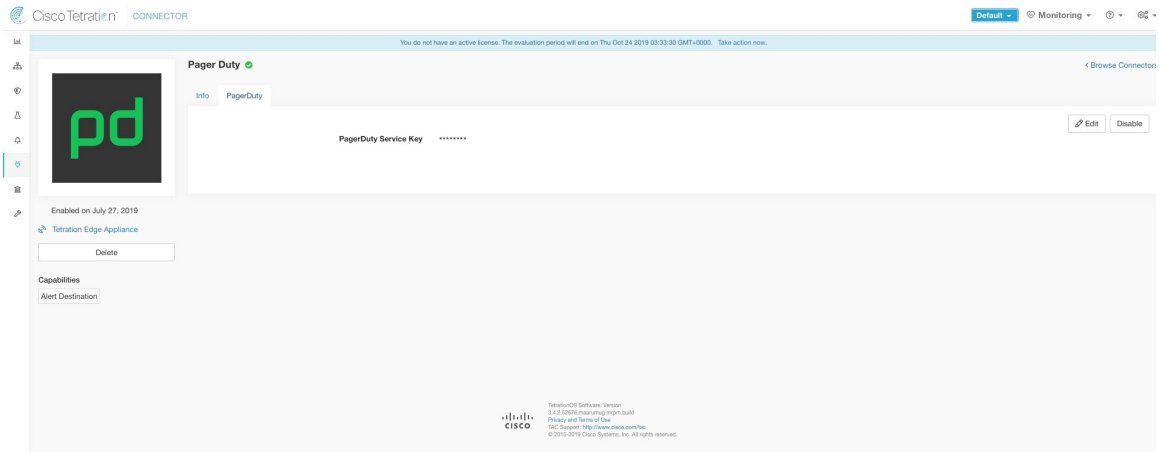
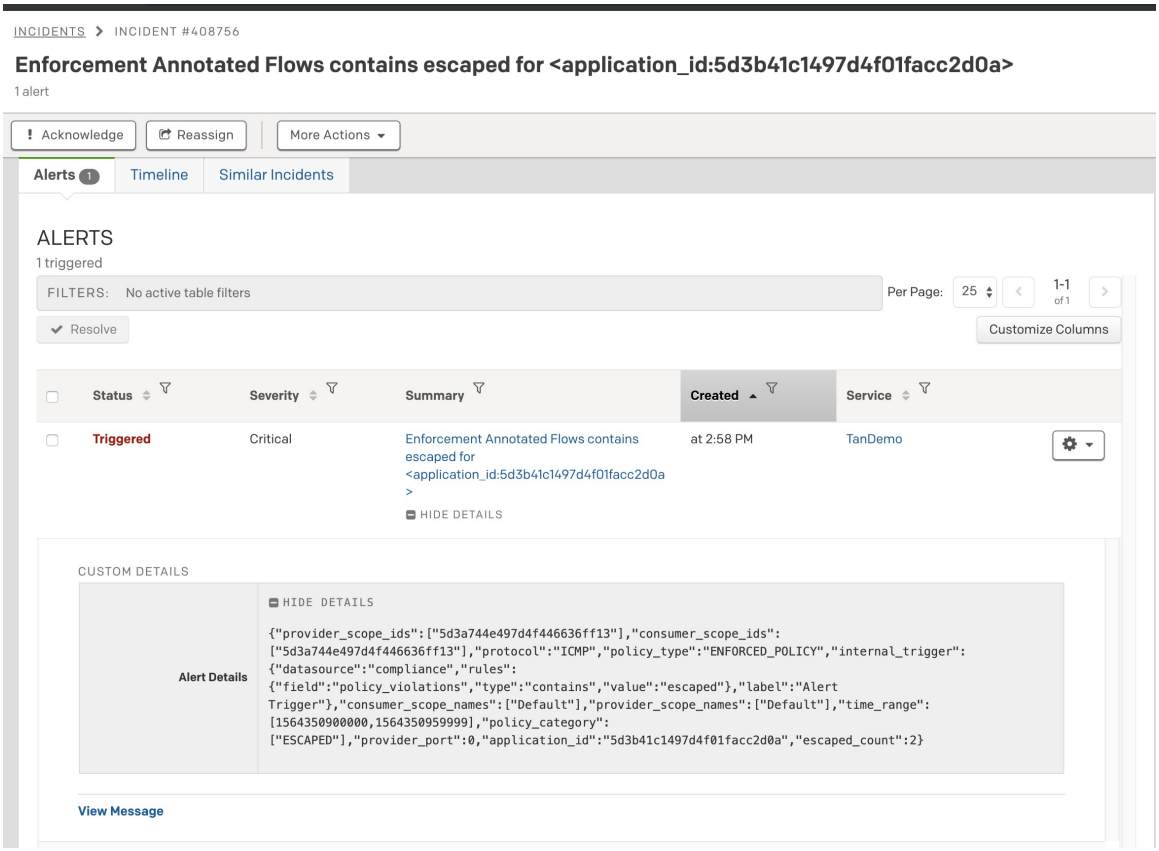


図 *92* : サンプルアラート



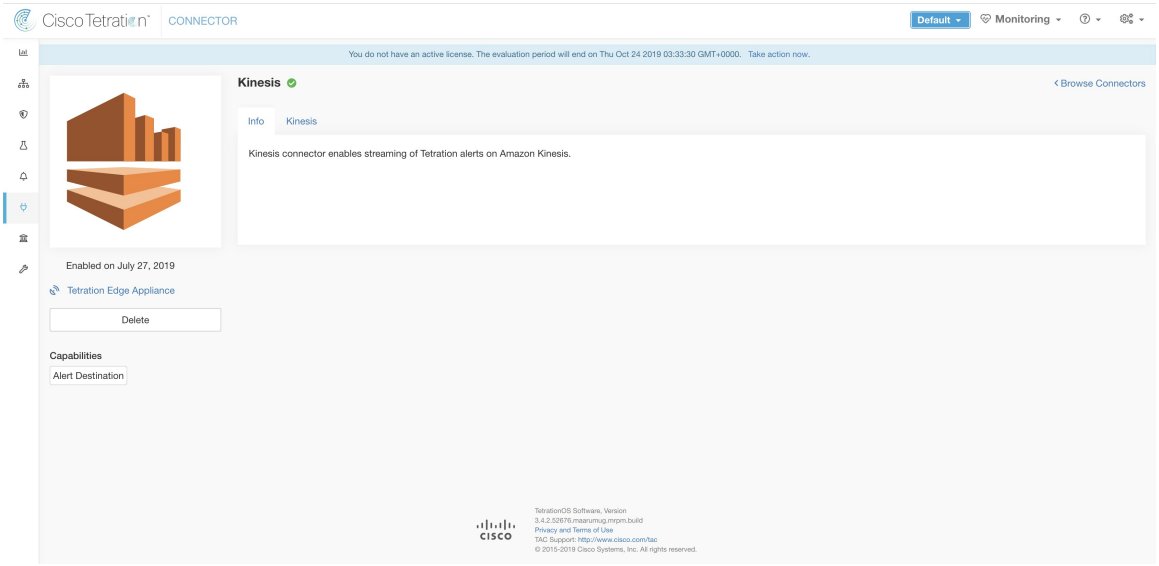
制限

メトリック	制限
1 つの Secure Workload Edge アプライアンスにおける PagerDuty コネクタの最大数	1
1 つのテナント（ルート範囲）における PagerDuty コネクタの最大数	1
Cisco Secure Workload における PagerDuty コネクタの最大数	150

Kinesis コネクタ

有効にすると、Secure Workload Edge アプライアンスの TAN サービスは、設定に基づいてアラートを送信できます。

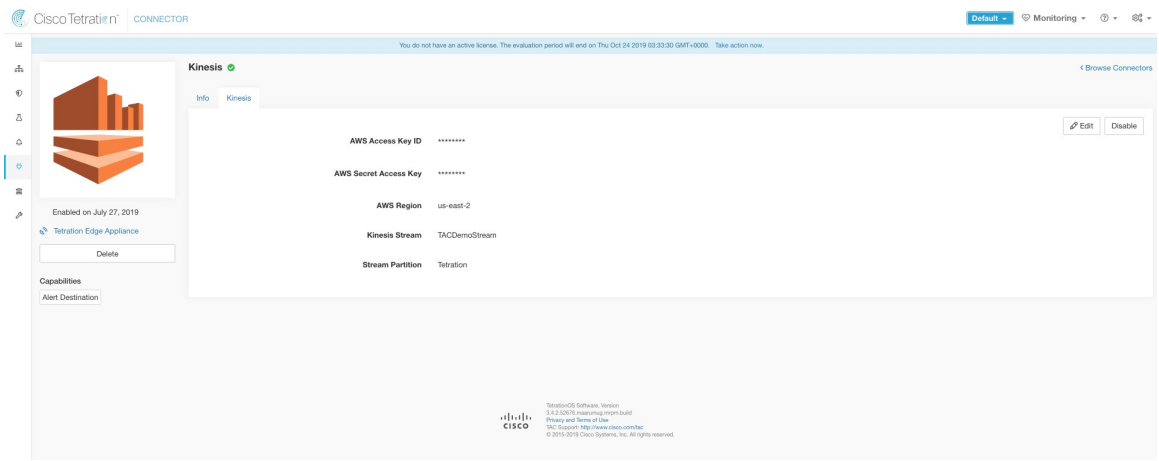
図 *93 : Kinesis* コネクタ



次の表は、Amazon Kinesis で Secure Workload アラートを公開するための設定の詳細を示しています。詳細については、「Kinesis 通知設定」を参照してください。

パラメータ名	タイプ	説明
AWS アクセスキー ID（AWS Access Key ID）	string	AWS と通信するための AWS アクセスキー ID
AWS 秘密アクセスキー（AWS Secret Access Key）	string	AWS と通信するための AWS シークレットアクセスキー
AWS リージョン（AWS Region）	AWS リージョンのドロップダウン	Kinesis ストリームが設定されている AWS リージョンの名前
Kinesis ストリーム（Kinesis Stream）	string	Kinesis ストリームの名前
ストリームパーティション（Stream Partition）	string	ストリームのパーティション名

図 *94 : Kinesis* コネクタの設定例



制限

メトリック	制限
1 つの Secure Workload Edge アプライアンスにおける Kinesis コネクタの最大数	1
1 つのテナント（ルート範囲）における Kinesis コネクタの最大数	1
Cisco Secure Workload における Kinesis コネクタの最大数	150

Cloud Connector

クラウドベースのワークロードでの Secure Workload 機能に、Cloud Connector を使用できます。

Cloud Connector には、仮想アプライアンスは必要ありません。

コネクタ	サポートされる機能	仮想アプライアンス上に展開
AWS	Amazon Web Services VPC 向け： • メタデータ（ラベル）の収集 • フローログの収集 • セグメンテーションポリシーの適用 Elastic Kubernetes Service（EKS）クラスタから： • メタデータの収集	該当なし
Azure	Azure VNet 向け： • メタデータ（ラベル）の収集 • フローログの収集 • セグメンテーションポリシーの適用 Azure Kubernetes Service（AKS）クラスタから： • メタデータの収集	該当なし
GCP	Google Cloud Platform VPC の場合： • メタデータ（ラベル）の収集 • フローログの収集 • セグメンテーションポリシーの適用 Google Kubernetes Engine（GKE）クラスタから： • メタデータ（ラベル）の収集	該当なし

AWS コネクタ

Amazon Web Services（AWS）コネクタは AWS に接続して、次の高レベルの機能を実行します。

- **AWS Virtual Private Cloud（VPC）からライブでのインベントリ（およびそのラベル）の自動取り込み** AWS では、タグの形式でリソースにメタデータを割り当てることができます。Secure Workload は、これらのリソースのタグについてクエリを実行します。これらのタグは、インベントリおよびトラフィックフローデータの視覚化、およびポリシー定義に使用できます。この機能では、このデータを常に同期することにより、リソースタグのマッピングが最新の状態に保たれます。

  タグは、AWS VPC のワークロードとネットワーク インターフェイスから取り込まれます。ワークロードとネットワークインターフェイスの両方が構成されている場合、タグはマージされて Cisco Secure Workload に表示されます。詳細については、クラウドコネクタによって生成されたラベル （410 ページ）を参照してください。

- **VPC レベルのフローログの取り込み** 監視目的で AWS で VPC フローログを設定している場合、Secure Workload は、対応する S3 バケットを読み取ることでフローログ情報を取り込むことができます。このテレメトリは、視覚化およびセグメンテーションポリシーの生成に使用できます。

- **セグメンテーション** このオプションを有効にすると、Secure Workload は、AWS のネイティブセキュリティグループを使用してセキュリティポリシーをプログラムします。VPC に対して適用が有効になっている場合、関連するポリシーがセキュリティグループとして自動的にプログラムされます。

- **EKS クラスタからのメタデータの自動取り込み** AWS で Elastic Kubernetes Services（EKS）が実行されている場合、選択したすべての Kubernetes クラスタに関連するすべてのノード、サービス、およびポッドのメタデータを収集することを選択できます。

VPC ごとに有効にする機能を選択できます。

（注） 中国地域は、現在サポートされていません。

AWS の要件と前提条件

すべての機能に対して：AWS で専用ユーザーを作成するか、このコネクタの既存の AWS ユーザーを特定します。コネクタ構成ウィザードは、このユーザーに必要な権限を割り当てるために使用できる CloudFormation テンプレート（CFT）を生成します。この CFT をアップロードするためのアクセス許可が AWS にあることを確認してください。

専用ユーザーにクロス AWS アカウントアクセスを許可する方法については、以下の cross_account セクションを参照してください。必要なアクセス権限についても説明しています。

ロールを使用して AWS アカウントにアクセス権を付与するには、Secure Workload クラスタへのロールベースのアクセスを参照してください。

各 VPC は、1 つの AWS コネクタのみに属することができます。1 つの Secure Workload クラスタは、複数の AWS コネクタを持つことができます。以下の「AWS コネクタの設定新しい AWS コネクタの設定 （295 ページ）」の表で説明されている情報を収集します。

このコネクタには、仮想アプライアンスは必要ありません。

ラベルとインベントリを収集するには：追加の前提条件は必要ありません。

フローログを取り込むには：フローログの収集をトリガーするには、VPC レベルのフローログ定義が必要です。

VPC レベルのフローログのみを取り込むことができます。

フローログは Amazon Simple Storage Service（S3）に発行する必要があります。Secure Workload は、Amazon CloudWatch ログからフローデータを収集できません。

コネクタの作成中に提供された AWS ユーザーアカウントのログイン情報で VPC フローログと S3 バケットの両方にアクセスできる場合、Cisco Secure Workload は、任意のアカウントに関連付けられた S3 バケットからフローログを取り込むことができます。

フローログには、次のフローログ属性（順序は任意）が必要です。送信元アドレス、宛先アドレス、送信元ポート、宛先ポート、プロトコル、パケット数、バイト数、開始時刻、終了時刻、アクション、TCP フラグ、Interface-ID、ログのステータス、フローの方向。その他の属性は無視されます。

フローログは、許可されたトラフィックと拒否されたトラフィックの両方をキャプチャする必要があります。

（注） Secure Workload AWS コネクタは、時間単位の VPC ログパーティションをサポートしていません。

セグメンテーション：セグメンテーションを有効にするには、ラベルの収集を有効にする必要があります。

VPC のセグメンテーションを有効にすると、既存のすべてのルールが上書きされるため、コネクタでセグメンテーションを有効にする前に、既存のセキュリティグループをバックアップしてください。

以下の「AWS インベントリにセグメンテーションポリシーを適用するときのベストプラクティス」も参照してください。

マネージド Kubernetes サービス（EKS）：Kubernetes オプションを有効にする場合は、必要なアクセス権限を含む、以下の AWS（EKS）で実行されるマネージド Kubernetes サービスのセクションで要件と前提条件を参照してください。

（オプション）AWS でクロス AWS アカウントアクセスを設定する

指定されたユーザーログイン情報で他の AWS アカウントに属する VPC にアクセスできる場合、それらは AWS コネクタの一部として処理できるようになります。

1. 指定された Secure Workload ユーザーには、次の AWS アクセス許可が必要です。

```
1. iam:GetPolicyVersion
2. iam:ListPolicyVersions
3. iam:ListAttachedUserPolicies
4. iam:GetUser
```

AWS ポリシー JSON の例：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "iam:GetPolicyVersion",
        "iam:ListPolicyVersions",
        "iam:ListAttachedUserPolicies",
        "iam:GetUser"
      ],
      "Resource": "*"
    }
  ]
}
```

2. 指定された Secure Workload ユーザーが属していない目的の AWS アカウントに AWS IAM ロールを作成します。

3. Secure Workload ユーザーが AWS IAM ロールを引き受けることを許可します。これは、AWS IAM ロールの信頼ポリシーに Secure Workload ユーザー ARN を追加することで実行できます。

   AWS IAM ロールの信頼ポリシー JSON の例：

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "AWS": <Secure Workload_user_arn>
            },
  "Action": "sts:AssumeRole",
  "Condition": {}
    }
  ]
}
```

4. Secure Workload ユーザーが属していないすべての目的の AWS アカウントに対して、手順 2 と 3 を実行します。

5. 異なるアカウントから作成されたすべての AWS ロールを引き受ける権限を持つカスタマー管理ポリシー（インラインポリシーではない）を作成します。

（注） AWS コネクタでは、カスタマー インライン ポリシーはサポートされていません。

管理ポリシー JSON の例：

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
     "Resource": [<AWS_role_cross_account_1_arn>, <AWS_role_cross_account_2_arn>...]

    }
  ]
}
```

6. 作成したカスタマー管理ポリシーを Secure Workload ユーザーにアタッチします。
7. コネクタ構成ウィザードは CloudFormation テンプレートを提供します。指定された Secure Workload ユーザーに CFT をそのままアップロードした後、テンプレートを編集し、編集したテンプレートを CloudFormation ポータルにアップロードして、AWS IAM ロールに必要な権限を付与します。詳細については、「AWS コネクタの設定新しい AWS コネクタの設定 （295 ページ）」を参照してください。

ロールを使用した認証

ユーザーベースの認証には、クレデンシャルキーが必要です。クレデンシャルキーが適切に管理されていないと、その機密性により、セキュリティ脅威が発生する可能性があります。

ロールベース認証を使用すると、ロールを使用して AWS アカウントを設定できます。コネクタ設定はロール ID（ARN）を受け入れ、そのロールを引き受けてお客様のアカウントで特定のアクションを実行します。

ロールベース認証により、不正アクセスのリスクが軽減されます。

ロールベース認証にアクセスするには、次の手順を実行します。

手順

ステップ 1 コネクタ設定ページの [ロール（Role）] タブをクリックします。

ステップ 2 クラスタを登録します。クラスタが登録されていない場合は、「ロールのログイン情報を使用するためにクラスタが登録されていません。提供されたペイロードをダウンロードして、**TAC** チームにお問い合わせください（Cluster is not registered to use Role credentials. Please download the provided payload and contact the TAC team）」という通知メッセージが表示されます。

ステップ 3 通知メッセージから [ダウンロード（download）] ボタンをクリックして、ペイロードファイルをダウンロードします。

ステップ 4 通知メッセージ内のリンクを使用して **TAC** チームに問い合わせ、チケットを発行し、ダウンロードしたファイルを提供することができます。

ステップ 5 クラスタが登録されると、[外部ID（External Id）] と [ユーザーARN（User ARN）] が自動的に入力されます。

（注） ページを更新して、[外部ID（External Id）] と [ユーザーARN（User ARN）] を表示します。

ステップ 6 生成された [外部ID（External Id）] と [ユーザーARN（User ARN）] を使用して、ロールの信頼関係を更新します。これにより、ロールの引き受けが可能になります。

JSON ファイルの同じ部分を以下に示します。

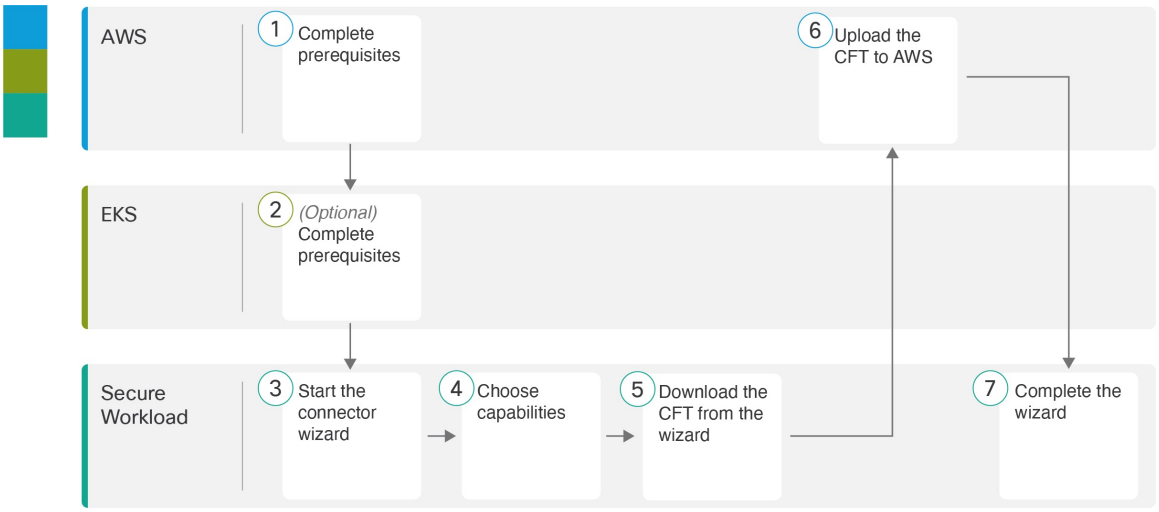
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": “<User ARN>"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": “<External Id>"
        }
      }
    }
  ]
}
```

ステップ 7 前の手順が完了したら、AWS アカウントから [ロールARN（Role ARN）] をコピーして、AWS コネクタの設定ページで貼り付けることができます。

AWS コネクタ設定の概要

次の図は、コネクタ設定プロセスの高度な概要を示しています。重要な詳細については、次のトピック（「AWS コネクタの設定新しい AWS コネクタの設定 （295 ページ）」）を参照してください。

図 *95 : AWS* コネクタ設定の概要



（図中の番号は、詳細手順のステップ番号に対応していないことに注意してください。）

新しい AWS コネクタの設定

手順

ステップ 1 ウィンドウの左側にあるナビゲーションバーから、**[管理（Manage）] > [ワークロード（Workloads）] > [コネクタ（Connectors）]** を選択します。

ステップ 2 該当する AWS コネクタをクリックします。

ステップ 3 [テンプレートの生成（Generate Template）] をクリックし、目的の機能を選択します。

選択した機能に基づいて、CloudFormation テンプレート（CFT）が生成されます。AWS CloudFormation で生成された CFT テンプレートを使用して、ユーザーやロールのポリシーを作成します。

セグメンテーションを有効にするには、[ラベルの収集（Gather Labels）] も有効にする必要があります。

ステップ 4 生成された CloudFormation テンプレート（CFT）をダウンロードします。生成された CFT は、ユーザーとロールの両方に使用できます。

このテンプレートには、前の手順で選択した機能に必要な IAM 権限があります。

Kubernetes オプションを有効にした場合は、EKS へのアクセス許可を別個に設定する必要があります。AWS（EKS）で実行されるマネージド Kubernetes サービス （305 ページ）を参照してください。

ステップ 5 CFT を AWS CloudFormation ポータルにアップロードして、このコネクタのユーザーに権限を割り当てます。AWS コネクタの設定に進む前に、AWS ユーザーに必要な権限があることを確認してください。

（注） これは、AWS クロスアカウントアクセスを使用しているかどうかに関係なく推奨されるタスクです。

ポータルまたは CLI を使用して CFT を適用できます。この説明については、以下を参照してください。

- **ポータル**：AWS 管理コンソール
- **CLI**：スタックの作成

CFT をアップロードすると、AWS は次の情報を要求します。

1. ポリシーの名前（任意の名前を指定できます。例：Secure Workload Connector）
2. ロール名：CFT を適用する AWS IAM ロールの名前
3. バケット ARN とオブジェクト ARN のリスト（デフォルト：*）
4. ユーザー名：CFT を適用する AWS ユーザーの名前
5. VPC ARN のリスト（デフォルト：*）

ステップ 6 AWS ロールベース認証を使用して Secure Workload コネクタに接続する場合は、EKS ロールおよびアクセス権限 （306 ページ）を参照してください。

ステップ 7 AWS クロスアカウントアクセスを使用している場合の追加の手順：

1. アップロードされた同じ CFT を使用して、ロールやユーザーにアクセス権を付与することができます。複数のアカウントがある場合は、各アカウントで同じ CFT を使用します。
2. CFT を、目的の IAM ロールが存在する各 AWS アカウントの AWS CloudFormation ポータルにアップロードします。

   前の手順で説明したように、ポータルまたは CLI を使用して CFT を適用できます。

   CFT をアップロードすると、AWS は次の情報を要求します。

   1. ポリシーの名前（任意の名前を指定できます。例：Secure Workload Connector）
   2. バケット ARN とオブジェクト ARN のリスト（デフォルト：*）
   3. ロール名：CFT を適用する AWS IAM ロールの名前
   4. VPC ARN のリスト（デフォルト：*）

ステップ 8 [スタートアップガイド（Getting started guide）]（推奨）または [ここに新しいコネクタを設定する（Configure your new connector here）] ボタンをクリックして、コネクタを設定します。

ステップ 9 「AWS の要件と前提条件」、EKS ロールおよびアクセス権限 （306 ページ）、「AWS インベントリにセグメンテーションポリシーを適用するときのベストプラクティス」を理解して満たしてから、[開始する（Get Started）] をクリックします。または、[新しいコネクタを設定する（Configure your new connector）] ボタンを使用して設定する場合は、[はい（yes）] をクリックします。

ステップ 10 コネクタに名前を付けて、説明を入力します。

ステップ 11 設定を次のように構成します。

いずれかのオプションを使用して AWS アカウントに接続できます。

1. クレデンシャルキー（Credential Keys）
2. ロール（Roles）

パラメータ名	属性	説明
クレデンシャルキー（Credential Keys）	Access Key	上記の CFT で説明されている権限を持つAWSユーザーに関連付けられた ACCESS KEY ID。
	秘密キー（Secret Key）	上記の ACCESS KEY ID に関連付けられた SECRET KEY。
ロール（Roles）	外部Id（External Id）	AWS リソースへのアクセス権を付与するための、自動生成された固有識別子です。ユーザーがロールに信頼関係を追加するために使用されます。
	ユーザーARN（User ARN）	IAM に割り当てられる、自動生成された固有識別子です。ユーザーがロールに信頼関係を追加するために使用されます。
	ARN	各AWSリソースに割り当てられる固有識別子です。
	HTTP プロキシ	（オプション）AWS に到達するために Secure Workload に必要なプロキシです。
	フルスキャン間隔	Secure Workload が AWS からの完全なインベントリデータを更新する頻度。最小値は 3600 秒で、これがデフォルトです。
	差分スキャン間隔	Secure Workload が AWS からインベントリデータの増分変更情報を取得する頻度。最小値は 600 秒で、これがデフォルトです。

ステップ 12 [次へ（Next）] をクリックします。

ステップ 13 次のページには、ユーザーが展開してさまざまなリージョンを表示できる、**リソースツリー**が表示されます。リージョン内では、リソースのチェックボックスを選択または選択解除して、AWS から VPC および EKS クラスタのリストを取得することができます。

ステップ 14 VPC（仮想ネットワーク）のリストから、選択した機能を有効にする VPC を選択します。

一般に、Secure Workload が適正なポリシーを提案するのに必要な十分なデータの収集を開始できるように、できるだけ早くフローの取り込みを有効にする必要があります。

EKS はラベルの収集機能のみをサポートしているため、明示的な機能の選択はできないことに注意してください。EKS クラスタを選択すると、サポートされている機能が暗黙的に有効になります。この機能を有効にするクラスタごとに、**Assume Role ARN**（Secure Workload への接続中に引き受けるロールの Amazon リソース番号）を入力します。

VPC で [セグメンテーションの有効化（Enable Segmentation）] を使用すると、既存のセキュリティグループが削除され、すべての VPC にデフォルトアクセスが提供されます。

通常は、初期設定時に [セグメンテーションの有効化（Enable Segmentation）] を選択しないでください。後で、特定の VPC にセグメンテーションポリシーを適用する準備ができたら、コネクタを編集して、それらの VPC のセグメンテーションを有効にすることができます。AWS インベントリにセグメンテーションポリシーを適用する際のベストプラクティスを参照してください。

ステップ 15 EKS クラスタの場合、AWS コネクタに接続するための Assume Role ARN アクセス ID を指定することで、AWS IAM ロールアクセスを許可できます。

ステップ 16 選択が完了したら、[作成（Create）] をクリックし、検証チェックが完了するまで数分待ちます。

次のタスク

ラベルの収集、フローデータの取り込み、および/またはセグメンテーションを有効にしている場合：

- フローの取り込みを有効にした場合、フローが **[調査（Investigate）] > [トラフィック（Traffic）]** ページに表示されるまでに最大 25 分かかる場合があります。
- （オプション）より豊富なフローデータと、ホストの脆弱性の可視化（CVE）といったその他の利点を得るため、VPC ベースのワークロードにオペレーティングシステムに適切なエージェントをインストールします。要件と詳細については、エージェントのインストールの章を参照してください。
- ラベルを収集してフローを取り込むように AWS コネクタを正常に設定した後は、セグメンテーションポリシーを構築するための標準プロセスに従います。例：Secure Workload が以下を実行できるようにします。十分なフローデータを収集して信頼できるポリシーを生成する。スコープを定義または変更する（通常は VPC ごとに 1 つ）。範囲ごとにワークスペースを作成する。フローデータに基づいてポリシーを自動的に検出する、および/または手動でポリシーを作成する。ポリシーを分析して改善する。ポリシーが下記のガイドラインとベストプラクティスを満たしていることを確認する。準備ができたら、ワークス

ベースで該当するポリシーを承認して適用します。特定の VPC にセグメンテーションポリシーを適用する準備ができたら、コネクタ設定に戻って VPC のセグメンテーションを有効にします。詳細については、AWS インベントリにセグメンテーションポリシーを適用するときのベストプラクティス （304 ページ）を参照してください。

Kubernetes マネージドサービス（EKS）オプションを有効にしている場合：

- コンテナベースのワークロードにKubernetesエージェントをインストールします。詳細については、エージェント展開の章の「*Kubernetes/Openshift* エージェント：詳細可視性と適用」セクションを参照してください。

イベントログ：

イベントログを使用すると、さまざまな機能からコネクタごとに発生する重要なイベントを把握することができます。コンポーネント、名前空間、メッセージ、およびタイムスタンプなどのさまざまな属性を使用してフィルタできます。

AWS コネクタの設定

手順

ステップ 1 ウィンドウの左側にあるナビゲーションバーから、**[管理（Manage）]>[コネクタ（Connectors）]** を選択します。

ステップ 2 該当する AWS コネクタをクリックします。

ステップ 3 （ルート範囲内の）最初のコネクタの場合は [有効化（Enable）] をクリックし、同じルート範囲内の付加的なコネクタの場合は [別のコネクタを有効化（Enable Another）] をクリックします。

ステップ 4 AWS の要件と前提条件を理解し満たしてから、[開始する（Get Started）] をクリックします。

ステップ 5 コネクタに名前を付け、必要な機能を選択して、[次へ（Next）] をクリックします。

このページでの選択は、次の手順で生成される CloudFormation テンプレート（CFT）に含まれる権限を決定し、設定が必要な項目を表示するためにのみ使用されます。

セグメンテーションを有効にするには、[ラベルの収集（Gather Labels）] も有効にする必要があります。

このページでセグメンテーションを有効にしても、それだけでポリシーの適用が有効になったり、既存のセキュリティグループに影響が及んだりすることはありません。ポリシーの適用と既存のセキュリティグループの削除は、以後ウィザードで個々の VPC のセグメンテーションを有効にした場合にのみ発生します。後でこのウィザードに戻って、個々の VPC のセグメンテーションポリシーの適用を有効にすることができます。

ステップ 6 生成された CloudFormation テンプレート（CFT）をダウンロードします。

このテンプレートには、前の手順で選択した機能に必要な IAM 権限があります。

Kubernetes オプションを有効にした場合は、EKS へのアクセス許可を別個に設定する必要があります。下の「AWS（EKS）で実行されるマネージド Kubernetes サービス」セクションを参照してください。

ステップ 7 CFT を AWS CloudFormation ポータルにアップロードして、このコネクタのユーザーに権限を割り当てます。ウィザードで次のページに進む前に、AWS ユーザーに必要な権限があることを確認してください。

（注） これは、AWS クロスアカウントアクセスを使用しているかどうかに関係なく推奨されるタスクです。

ポータルまたは CLI を使用して CFT を適用できます。この説明については、以下を参照してください。

- **ポータル**：AWS 管理コンソール
- **CLI**：スタックの作成

CFT をアップロードすると、AWS は次の情報を要求します。

1. ポリシーの名前（任意の名前を指定できます。例：Secure Workload Connector）
2. バケット ARN とオブジェクト ARN のリスト（デフォルト：*）
3. ユーザー名：CFT を適用する AWS ユーザーの名前
4. VPC ARN のリスト（デフォルト：*）

ステップ 8 AWS クロスアカウントアクセスを使用している場合の追加の手順：

1. ウィザードからダウンロードした CloudFormation テンプレートを編集します。

   編集前のテンプレートの関連パーツ：

```
                },
                "Users": [
                    {
                        "Ref": "Username"
                    }
                ]
            }
        }
    },
    "Parameters": {
        "PolicyName": {
            "Type": "String",
            "Default": "",
            "Description": "Name of the policy. Example: CiscoSecureWorkloadPolicy"
        },
        "Username": {
            "Type": "String",
            "Default": "",
            "Description": "User name. Example: \"SecureWorkloadUser\""
        }
    }
}
```

   編集後の同じテンプレートパーツ：

2. 編集した CFT を、目的の IAM ロールが存在する各 AWS アカウントの AWS CloudFormation ポータルにアップロードします。

   前の手順で説明したように、ポータルまたは CLI を使用して CFT を適用できます。

   CFT をアップロードすると、AWS は次の情報を要求します。

   1. ポリシーの名前（任意の名前を指定できます。例：Secure Workload Connector）
   2. バケット ARN とオブジェクト ARN のリスト（デフォルト：*）
   3. ロール名：CFT を適用する AWS IAM ロールの名前
   4. VPC ARN のリスト（デフォルト：*）

ステップ 9 設定を次のように構成します。

属性	説明
Access Key	上記の CFT で説明されている権限を持つ AWS ユーザーに関連付けられた ACCESS KEY ID。
秘密キー（Secret Key）	上記の ACCESS KEY ID に関連付けられた SECRET KEY。
HTTP プロキシ	AWS に到達するために Secure Workload に必要なプロキシ。サポートされるプロキシポート：80、8080、443、および 3128。
フルスキャン間隔	Secure Workload が AWS からの完全なインベントリデータを更新する頻度。最小値は 3600 秒で、これがデフォルトです。
差分スキャン間隔	Secure Workload が AWS からインベントリデータの増分変更情報を取得する頻度。最小値は 600 秒で、これがデフォルトです。

ステップ 10 [次へ（Next）] をクリックします。システムが AWS から VPC と EKS クラスタのリストを取得するには、数分かかる場合があります。

ステップ 11 VPC（仮想ネットワーク）と、各 VPC の EKS クラスタのリストから、選択した機能を有効にする VPC と EKS クラスタを選択します。

一般に、Secure Workload が適正なポリシーを提案するのに必要な十分なデータの収集を開始できるように、できるだけ早くフローの取り込みを有効にする必要があります。

EKS はラベルの収集機能のみをサポートしているため、明示的な機能の選択はできないことに注意してください。EKS クラスタを選択すると、サポートされている機能が暗黙的に有効になります。この機能を有効にするクラスタごとに、**Assume Role ARN**（Secure Workload への接続中に引き受けるロールの Amazon リソース番号）を入力します。

通常は、初期設定時に [セグメンテーションの有効化（Enable Segmentation）] を選択しないでください。後で、特定の VPC にセグメンテーションポリシーを適用する準備ができたら、コネクタを編集して、それらの VPC のセグメンテーションを有効にすることができます。AWS インベントリにセグメンテーションポリシーを適用する際のベストプラクティスを参照してください。

ステップ 12 選択が完了したら、[作成（Create）] をクリックし、検証チェックが完了するまで数分待ちます。

[グループの表示（View Groups）] ページには、前のページで機能を有効にしたすべての VPC が地域別にグループ化されて表示されます。各地域、および各地域の各 VPC は、新しい範囲です。

ステップ 13 新しい範囲のセットを追加する親範囲を選択します。いずれの範囲もまだ定義していない場合、唯一のオプションはデフォルトの範囲になります。

ステップ 14 ウィザードで行われたすべての設定（階層型範囲ツリーを含む）を受け入れるには、[保存（Save）] をクリックします。

階層型範囲ツリーを除くすべての設定を受け入れるには、[この手順をスキップ（Skip this step）] をクリックします。

範囲ツリーは、後で **[整理（Organize）] > [範囲とインベントリ（Scopes and Inventory）]** で手動で作成または編集できます。

次のタスク

ラベルの収集、フローデータの取り込み、および/またはセグメンテーションを有効にしている場合：

- フローの取り込みを有効にした場合、フローが **[調査（Investigate）] > [トラフィック（Traffic）]** ページに表示されるまでに最大 25 分かかる場合があります。
- （オプション）より豊富なフローデータと、ホストの脆弱性の可視化（CVE）といったその他の利点を得るため、VPC ベースのワークロードにオペレーティングシステムに適切なエージェントをインストールします。要件と詳細については、エージェントのインストールの章を参照してください。
- ラベルを収集してフローを取り込むように AWS コネクタを正常に設定した後は、セグメンテーションポリシーを構築するための標準プロセスに従います。例：Secure Workload が以下を実行できるようにします。十分なフローデータを収集して信頼できるポリシーを生成する。スコープを定義または変更する（通常は VPC ごとに 1 つ）。範囲ごとにワークスペースを作成する。フローデータに基づいてポリシーを自動的に検出する、および/または手動でポリシーを作成する。ポリシーを分析して改善する。ポリシーが下記のガイドラインとベストプラクティスを満たしていることを確認する。準備ができたら、ワークスペースで該当するポリシーを承認して適用します。特定の VPC にセグメンテーションポリシーを適用する準備ができたら、コネクタ設定に戻って VPC のセグメンテーションを

有効にします。詳細については、AWS インベントリにセグメンテーションポリシーを適用するときのベストプラクティス （304 ページ）を参照してください。

Kubernetes マネージドサービス（EKS）オプションを有効にしている場合：

- コンテナベースのワークロードにKubernetesエージェントをインストールします。詳細については、エージェント展開の章の「*Kubernetes/Openshift* エージェント：詳細可視性と適用」セクションを参照してください。

AWS コネクタの編集

AWS コネクタを編集して、特定の VPC のセグメンテーションの適用を有効にしたり、他の変更を加えたりできます。

ウィザードを終了するまで、変更は保存されません。

手順

ステップ 1 ウィンドウの左側にあるナビゲーションバーから、**[管理（Manage）] > [ワークロード（Workloads）] > [コネクタ（Connectors）]** を選択します。

ステップ 2 [AWS] をクリックします。

ステップ 3 AWS コネクタが複数ある場合は、編集するコネクタをウィンドウの上部から選択します。

ステップ 4 [コネクタの編集（Edit Connector）] をクリックします。

ステップ 5 ウィザードをもう一度クリックして、変更を実行します。これらの設定の詳細については、AWS コネクタの設定新しい AWS コネクタの設定 （295 ページ）を参照してください。

ステップ 6 さまざまな機能（ラベルの収集、フローの取り込み、セグメンテーションの適用、またはEKSデータの収集）を有効にする場合は、ウィザードを続行する前に、改訂されたCloudFormationテンプレート（CFT）をダウンロードして AWS にアップロードする必要があります。

ステップ 7 セグメンテーションポリシーの適用を有効にするには、AWSインベントリにセグメンテーションポリシーを適用するときのベストプラクティスで、推奨される前提条件を満たしていることを最初に確認してください。VPC の一覧が表示されているページで、適用を有効にする VPC の [セグメンテーションの有効化（Enable Segmentation）] を選択します。

ステップ 8 ウィザードを使用するかまたは手動で、選択したいずれかの VPC の範囲をすでに作成している場合は、[この手順をスキップ（Skip this step）] をクリックしてウィザードを完了します。

[整理（Organize）] > [範囲とインベントリ（Scopes and Inventory）] ページを使用すると、範囲ツリーを手動で編集できます。

ステップ 9 選択した VPC の範囲をまだ作成しておらず、提案された階層を保持する場合は、範囲ツリーの上から親範囲を選択し、[保存（Save）] をクリックします。

コネクタとデータの削除

コネクタを削除しても、そのコネクタによってすでに取り込まれたデータは削除されません。

ラベルとインベントリは、24時間後にアクティブなインベントリから自動的に削除されます。

AWS インベントリにセグメンテーションポリシーを適用するときのベストプラクティス



警告 VPCでセグメンテーションの適用を有効にする前に、そのVPCでセキュリティグループのバックアップを作成します。VPC のセグメンテーションを有効にすると、その VPC から既存のセキュリティグループが削除されます。セグメンテーションを無効にしても、古いセキュリティグループは復元されません。

ポリシーの作成時：

- 検出されたすべてのポリシーと同様に、正確なポリシーを生成するための十分なフローデータがあることを確認してください。
- AWSはセキュリティグループではALLOWルールのみを許可するため、セグメンテーションポリシーには、Deny アクションを持つ必要がある Catch-All ポリシーを除き、Allow ポリシーのみを含める必要があります。

関連付けられた VPC のセグメンテーションを有効にする前に、ワークスペースで適用を有効にすることを推奨します。適用が有効になっているワークスペースに含まれていない VPC のセグメンテーションを有効にすると、その VPC ですべてのトラフィックが許可されます。

VPC にポリシーを適用する準備ができたら、AWS コネクタを編集し（「AWS コネクタの編集」を参照）、その VPC のセグメンテーションを有効にします。

AWS インベントリラベル、詳細、および適用ステータスの表示

AWS コネクタの概要情報を表示するには、コネクタページ（[管理（Manage）] > [コネクタ（Connectors）]）に移動し、ページの上部からコネクタを選択します。詳細については、[VPC] 行をクリックしてください。

AWS VPC インベントリに関する情報を表示するには、[AWSコネクタ（AWS Connectors）] ページで IP アドレスをクリックして、そのワークロードの [インベントリプロファイル（Inventory Profile）]ページを表示します。インベントリプロファイルの詳細については、「インベントリプロファイル」を参照してください。

ラベルの詳細については、次を参照してください。

- クラウドコネクタによって生成されたラベル
- Kubernetes クラスタに関連するラベル

VPC インベントリの具体的なポリシーは、その orchestrator_system/interface_id label ラベル値に基づいて生成されます。これは、[インベントリプロファイル（Inventory Profile）] ページで確認できます。

適用ステータスを表示するには、Secure Workload ウィンドウの左側のナビゲーションバーから **[保護（Defend）]** > **[適用ステータス（Enforcement Status）]** を選択します。詳細については、「Cloud Connector の適用ステータス」を参照してください。

AWS コネクタに関する問題のトラブルシューティング

問題：[適用ステータス（Enforcement Status）] ページに、具体的ポリシーがスキップされたと表示されます。

解決策：AWS コネクタで設定されているように、セキュリティグループの数が AWS の制限を超えると発生します。

具体的ポリシーが SKIPPED と表示されている場合、新しいセキュリティグループは実装されておらず、AWS に以前から存在していたセキュリティグループが引き続き有効です。

この問題を解決するには、小さなサブネットを持つ複数のポリシーではなく、1 つのポリシーで大きなサブネットを使用するなどして、ポリシーを統合できるかどうかを確認します。

ルール数の制限を増やすことを選択した場合は、AWS コネクタ設定の制限を変更する前に Amazon に連絡する必要があります。

バックグラウンド：

セグメンテーションを有効にすると、VPC ごとに具体的ポリシーが生成されます。生成された具体的ポリシーは、AWS でセキュリティグループを作成するために使用されます。ただし、AWS と Secure Workload ではポリシーのカウントが異なります。Secure Workload ポリシーを AWS セキュリティグループに変換する場合、AWS はそれぞれ一意のサブネットを 1 つのルールとしてカウントします。

アカウンティングの例：

次の Secure Workload ポリシーの例について考えてみます。

OUTBOUND: Consumer Address Set -> Provider Address Set Allow TCP port 80, 8080

AWS は、このポリシーを、「プロバイダーアドレスセット内の一意のサブネットの数」に「一意のポートの数」 を掛けたものとしてカウントします。

したがって、プロバイダーアドレスセットが 20 個の一意のサブネットで構成されている場合、この単一の Secure Workload ポリシーは、AWS で「20（一意のサブネット数）X 2（一意のポート数）= セキュリティグループ内に 40 のルール」としてカウントされます。

VPC が動的であるため、ルールカウントも動的です。カウントは概算であることに注意してください。

問題：AWS が予期せずすべてのトラフィックを許可する

対処方法：Secure Workload のキャッチオールポリシーが [拒否（Deny）] に設定されていることを確認します。

AWS（EKS）で実行されるマネージド Kubernetes サービス

AWS クラウドに Amazon Elastic Kubernetes Service（EKS）を展開した場合は、AWS コネクタを使用して、Kubernetes クラスタからインベントリとラベル（EKS タグ）をプルできます。

マネージド Kubernetes サービスからメタデータをプルするように AWS コネクタが設定されている場合、Secure Workload はクラスタの API サーバーに接続して、そのクラスタ内のノード、ポッド、およびサービスの状態を追跡します。このコネクタを使用して収集および生成された Kubernetes ラベルについては、「Kubernetes クラスタに関連するラベル」を参照してください。

EKS の要件および前提条件

- 使用している Kubernetes バージョンがサポートされていることを確認します。https://www.cisco.com/go/secure-workload/requirements/integrationsを参照してください。
- 以下の説明に従い、EKS で必要なアクセスを構成します。

EKS ロールおよびアクセス権限

ユーザーログイン情報と AssumeRole（該当する場合）は、最小限の権限セットで設定する必要があります。ユーザー/ロールは、aws-auth.yaml 設定マップで指定する必要があります。aws-auth.yaml 設定マップは、次のコマンドを使用して編集できます。

```
$ kubectl edit configmap -n kube-system aws-auth
```

AssumeRole が使用されていない場合、適切なグループを使用して aws-auth.yaml 設定マップの「mapUsers」セクションにユーザーを追加する必要があります。AssumeRole ARN が指定されている場合、aws-auth.yaml 設定マップの「mapRoles」セクションにロールを追加する必要があります。AssumeRole を使用した aws-auth.yaml 設定マップの例を以下に示します。

```
    apiVersion: v1
    data:
      mapAccounts: |
        []
      mapRoles: |
        - "groups":
          - "system:bootstrappers"
          - "system:nodes"
          "rolearn":
"arn:aws:iam::938996165657:role/eks-cluster-20210114181445234700000000a"
          "username": "system:node:{{EC2PrivateDNSName}}"
        - "rolearn": arn:aws:iam::938996165657:role/BasicPrivilegesRole
          "username": secure.workload.read.only-user
          "groups":
            - secure.workload.read.only

      mapUsers: |
        []
    kind: ConfigMap
    metadata:
      creationTimestamp: "2021-01-14T18:14:47Z"
      managedFields:
      - apiVersion: v1
        fieldsType: FieldsV1
        fieldsV1:
          f:data:
            .: {}
            f:mapAccounts: {}
            f:mapRoles: {}
            f:mapUsers: {}
        manager: HashiCorp
        operation: Update
        time: "2021-01-14T18:14:47Z"
      name: aws-auth
      namespace: kube-system
      resourceVersion: "829"
      selfLink: /api/v1/namespaces/kube-system/configmaps/aws-auth
      uid: 6c5a3ac7-58c7-4c57-a9c9-cad701110569
```

AWS コネクタウィザードでの *EKS* の設定

AWS コネクタを設定するときに、マネージド Kubernetes サービスの機能を有効にします。「AWS コネクタの設定新しい AWS コネクタの設定 （295 ページ）」を参照してください。

EKS クラスタごとに Assume Role ARN が必要になります。詳細については、以下を参照してください。 https://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html

AWS ユーザーを使用して EKS クラスタにアクセスしている場合は、そのユーザーに Assume Role へのアクセスを許可します。

クロスアカウントの IAM ロールを使用している場合は、IAM ロールが Assume Role にアクセスすることを許可します。

EKS ロードバランサのサポート

EKS で、ロードバランササービスのサポートが追加されました。CSW エージェントは、コンシューマホストとプロバイダーホスト/ポッドにルールを適用します。

EKS ロードバランサには、次の 2 つのオプションがあります。

1. [クライアントIPの保持（Preserve Client IP）]。
2. [ターゲットタイプ（Target Type）]。

ケースを開始する前、次のポリシーインテントの場合：

次のように生成された、さまざまなケースの許可アクションルールを持つ、コンシューマからプロバイダーへのサービス、サービスプロトコル、およびポート：

ケース	クライアントの保持	ターゲット タイプ（Target Type）
1	オン	IP
2	オン	インスタンス
3	オフ	IP
4	オフ	インスタンス

ケース 1：

コンシューマノードで、コンシューマからロードバランササービス（lb入力ip）へのサービスプロトコルとポートの許可を使用して出力ルールを生成します。

プロバイダーノードにホストルールはありませんが、送信元をコンシューマ、宛先をプロバイダーポッド（任意）、プロトコルをターゲットプロトコル、ポートをターゲットポート、アクションを許可として、プロバイダーポッドに入力ルールを生成します。

ケース 2：

コンシューマノードで、コンシューマからロードバランササービス（lb入力ip）へのサービスプロトコルとポートの許可を使用して出力ルールを生成します。

プロバイダーノードには、送信元をコンシューマ、宛先をすべてのプロバイダーノード、プロトコルをサービスプロトコル、ポートをサービスのノードポート、アクションを許可として生成された、事前ルーティングルールがあります。

プロバイダーポッドでは、送信元をプロバイダーノード、宛先をプロバイダーポッド（任意）、プロトコルをターゲットプロトコル、ポートをターゲットポート、アクションを許可として、入力ルールを生成します。

ケース 3：

コンシューマノードで、コンシューマからロードバランササービス（lb 入力 ip）へのサービスプロトコルとポートの許可を使用して出力ルールを生成します。

プロバイダーノードにホストルールはありません。プロバイダーポッドでは、送信元を lb 入力 ip、宛先をプロバイダーポッド（任意）、プロトコルをターゲットプロトコル、ポートをターゲットポート、アクションを許可として、入力ルールを生成します。

ケース 4：

コンシューマノードで、コンシューマからロードバランササービス（lb 入力 ip）へのサービスプロトコルとポートの許可を使用して出力ルールを生成します。

プロバイダーノードは、lb 入力 IP を送信元として設定し、すべてのプロバイダーノードを宛先として設定する、事前ルーティングルールを生成します。このルールでは、サービスプロトコルをプロトコルとして指定し、サービスのノードポートをポートとして指定し、アクションを許可に設定します。

プロバイダーポッドでは、送信元をプロバイダーノード、宛先をプロバイダーポッド（任意）、プロトコルをターゲットプロトコル、ポートをターゲットポート、アクションを許可として、入力ルールを生成します。

Azure コネクタ

Azure コネクタは、Microsoft Azure アカウントに接続して、次の高度な機能を実行します。

- **Azure 仮想ネットワーク（VNet）からのライブのインベントリ（およびそのタグ）の自動取り込み**：Azure では、タグ形式でリソースにメタデータを割り当てることができます。Secure Workload は仮想マシンとネットワーク インターフェイスに関連付けられたタグを取り込むことができます。取り込んだタグは、インベントリおよびトラフィックフローデータの可視化とポリシー定義のラベルとして Secure Workload で使用できます。このメタデータは常に同期されます。

  コネクタに関連付けられたサブスクリプションのワークロードとネットワーク インターフェイスからのタグが取り込まれます。ワークロードとネットワークインターフェイスの両方が構成されている場合、タグはマージされて Cisco Secure Workload に表示されます。詳細については、クラウドコネクタによって生成されたラベル（410 ページ）を参照してください。

- **フローログの取り込み**：コネクタは、Azure でネットワーク セキュリティ グループ（NSG）用に設定したフローログを取り込むことができます。その後、このテレメトリデータを Secure Workload で使用して、可視化およびセグメンテーションポリシーを生成できます。

- **セグメンテーション**：仮想ネットワークに対してセグメンテーションポリシーの適用が有効になっている場合、Secure Workload ポリシーは Azure のネイティブ ネットワーク セキュリティ グループを使用して適用されます。
- **AKS クラスタからのメタデータの自動取り込み**：Azure Kubernetes Services（AKS）が Azure で実行されている場合、選択したすべての Kubernetes クラスタに関連するすべてのノード、サービス、およびポッドのメタデータを収集できます。

上記の機能の中で VNet ごとに有効にする機能を選択できます。

Azure コネクタは、複数のサブスクリプションをサポートしています。

（注） 中国地域は、現在サポートされていません。

Azure の要件および前提条件

すべての機能：単一のコネクタで複数のサブスクリプションを処理できます。コネクタを設定するには、サブスクリプション ID が必要です。このサブスクリプション ID は、コネクタにオンボードされている多くのサブスクリプション ID のうちの 1 つとすることができます。

Azure で、Azure Active Directory（AD）を使用してアプリケーションを作成/登録します。このアプリケーションから、次の情報が必要になります。

- アプリケーション（クライアント）ID
- ディレクトリ（テナント）ID
- クライアント資格情報（証明書またはクライアント シークレットのいずれかを使用できます）
- サブスクリプション ID

コネクタ設定ウィザードは Azure Resource Manager（ARM）テンプレートを生成します。このテンプレートを使用して、有効化することを選択したコネクタ機能に必要なアクセス許可を持つカスタムロールを作成することができます。これらのアクセス許可は、コネクタに指定したサブスクリプション内のすべてのリソースに適用されます。このテンプレートをアップロードするためのアクセス許可が Azure にあることを確認してください。

接続に必要な場合は、この統合に使用できる HTTP プロキシがあることを確認してください。

各仮想ネットワーク（VNet）は、1 つの Azure コネクタにのみ属することができます。1 つの Azure アカウントに複数の Azure コネクタを設定することはできます。

このコネクタには、仮想アプライアンスは必要ありません。

ラベルとインベントリを収集するには：追加の前提条件は必要ありません。

フローログを取り込むには：各仮想ネットワーク（VNet）には、少なくとも 1 つのサブネットが設定されている必要があります。

各 VNet の下のすべてのサブネットには、ネットワーク セキュリティ グループ（NSG）が関連付けられている必要があります。1 つの NSG を複数のサブネットに関連付けることができます。NSG を設定するときに、任意のリソースグループを指定できます。

NSG ルールに一致するトラフィックのみがフローログに含まれます。したがって、すべての NSG には、インバウンド トラフィックとアウトバウンド トラフィックごとに少なくとも 1 つのルールが必要です。これらはすべての送信元、すべての宛先に適用され、Cisco Secure Workload のキャッチオールルールと同等です（デフォルトでは、NSG にはこれらのルールが含まれています）。

各 NSG でフローログが有効になっている必要があります。

- Azure のストレージアカウントが必要です。このコネクタに使用しているサブスクリプションにアクセス許可が含まれている必要があります。
- フローログはバージョン 2 を使用する必要があります。
- 保持期間は2 日間です（コネクタは毎分新しいフローデータをプルします。2 日間あれば、接続エラーを修正するのに十分な時間が取れます）。

セグメンテーション：セグメンテーションを有効にするには、ラベルの収集を有効にする必要があります。

仮想ネットワーク（VNet）のセグメンテーションを有効にすると、既存のすべてのルールが、サブネットに関連付けられた NSG と、それらのサブネットの一部であるネットワーク インターフェイスから削除されます。コネクタでセグメンテーションを有効にする前に、サブネットとネットワーク インターフェイスの既存の NSG ルールをバックアップしてください。

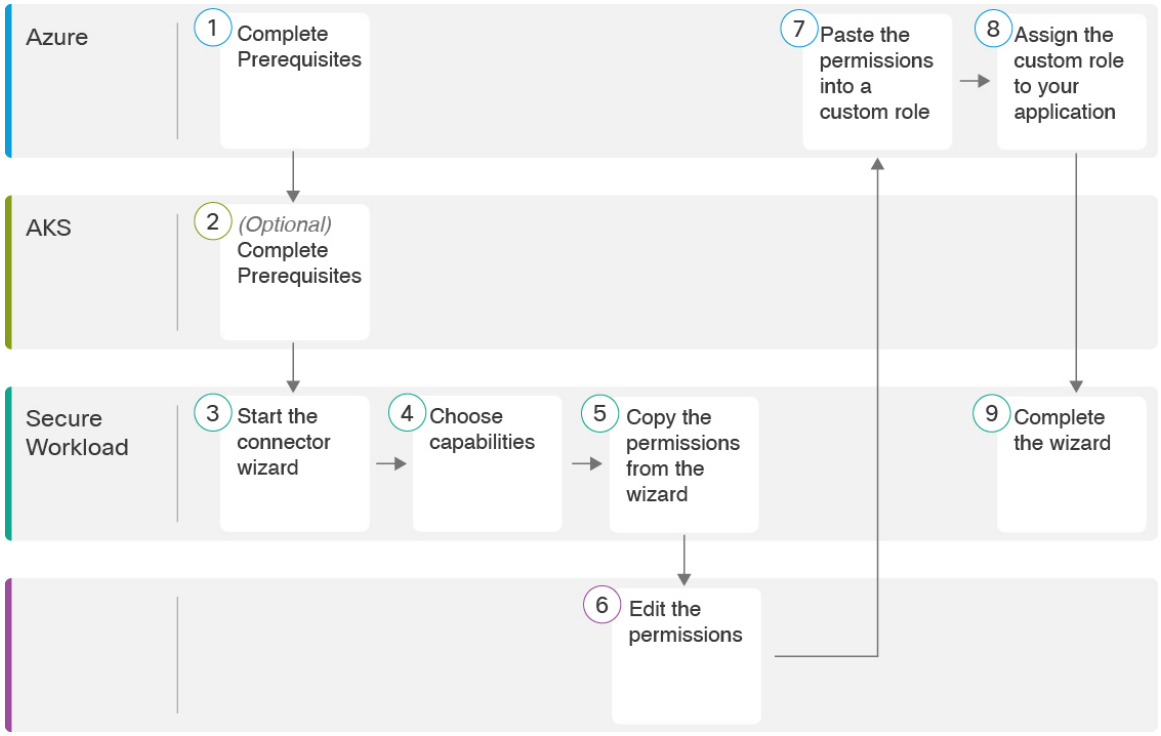
以下のAzure Inventory にセグメンテーションポリシーを適用するときのベストプラクティス（316 ページ）も参照してください。

マネージド Kubernetes サービス（AKS）：Kubernetes AKS オプションを有効にする場合は、以下の Azure（AKS）で実行されるマネージド Kubernetes サービスのセクションで要件と前提条件を参照してください。

Azure コネクタ構成の概要

次の図は、コネクタ構成プロセスの概要を示しています。重要な詳細については、次のトピック（「Azure コネクタの設定」）を参照してください。

図 *96 : Azure* コネクタ構成の概要



（図中の番号は、詳細手順のステップ番号に対応していないことに注意してください。）

Azure コネクタの設定

手順

ステップ 1 ウィンドウの左側にあるナビゲーションバーから、**[管理（Manage）]>[コネクタ（Connectors）]** を選択します。

ステップ 2 該当する Azure コネクタをクリックします。

ステップ 3 （ルート範囲内の）最初のコネクタの場合は [有効化（Enable）] をクリックし、同じルート範囲内の付加的なコネクタの場合は [別のコネクタを有効化（Enable Another）] をクリックします。

ステップ 4 「Azure の要件および前提条件」の要件と前提条件を理解し満たしてから、[開始する（Get Started）] をクリックします。

ステップ 5 コネクタに名前を付け、必要な機能を選択します。

このページでの選択は、次の手順で生成される Azure Resource Manager（ARM）テンプレートに含まれる特権を決定し、設定が必要な項目を表示するためにのみ使用されます。

セグメンテーションを有効にするには、[ラベルの収集（Gather Labels）] も有効にする必要があります。

このページでセグメンテーションを有効にしても、それだけでポリシーの適用が有効になったり、既存のネットワーク セキュリティ グループに影響が及んだりすることはありません。ポリシーの適用と既存のセキュリティグループの削除は、以後ウィザードで個々の Vnet のセグメンテーションを有効にした場合にのみ発生します。後でこのウィザードに戻って、個々の VNet のセグメンテーションポリシーの適用を有効にすることができます。

ステップ 6 [次へ（Next）] をクリックして、設定ページの情報に目を通します。

ステップ 7 ウィザードの次のページに進むには、サブスクリプションが必須の権限を備えている必要があります。

提供される Azure Resource Manager（ARM）テンプレートを使用して、コネクタに必要なアクセス許可を割り当てるには、次の手順に従います。

1. ウィザードから ARM テンプレートをダウンロードします。
2. テンプレートテキストを編集して、**<subscription_ID>** をユーザーのサブスクリプション ID に置き換えます。

   （注） コネクタの場合、Azure アカウント内に複数のサブスクリプション ID を作成できます。

   複数のログイン情報が同じサブスクリプション ID に属する場合、複数のサブスクリプション ID を入力できます。

3. Azure で、該当するサブスクリプションにカスタムロールを作成します。
4. ベースラインアクセス許可に関するカスタムロールフォームで、[最初から開始（Start from scratch）] を選択します。
5. カスタムロール作成フォームの [JSON] タブで、コネクタウィザードからダウンロードした編集済みファイルのテキストを貼り付けます。
6. カスタムロールを保存します。
7. この手順の前提条件で設定したアプリケーションにカスタムロールを付与します。

このテンプレートには、前の手順で選択した機能に必要な IAM アクセス許可があります。

Kubernetes マネージド サービス オプションを有効にした場合は、AKS へのアクセス許可を個別に設定する必要があります。以下の「Azure（AKS）で実行されるマネージド Kubernetes サービス （317 ページ）」セクションを参照してください。

ステップ 8 設定を次のように構成します。

属性	説明
SubscriptionID	このコネクタに関連付ける Azure サブスクリプションの ID。
ClientID	このコネクタ用に Azure で作成したアプリケーションの**アプリケーション（クライアント）ID**。

属性	説明
TenantID	このコネクタ用にAzureで作成したアプリケーションの**ディレクトリ（テナント）ID**。
クライアントシークレットまたはクライアント証明書	認証には、クライアントシークレットまたはクライアント証明書とキーのいずれかを使用できます。どちらも、このコネクタ用にAzureで作成したアプリケーションの**クライアントログイン情報**リンクから取得します。証明書を使用する場合：暗号化されていない証明書を使用する必要があります。RSA 証明書のみがサポートされます。秘密キーは、PKCS1 と PKCS8 のいずれかを使用できます。
HTTP プロキシ	Azure に到達するために Secure Workload に必要なプロキシ。サポートされるプロキシポート：80、8080、443、および 3128。
フルスキャン間隔	Secure Workload が Azure からの完全なインベントリデータを更新する頻度。最小値は 3600 秒で、これがデフォルトです。
差分スキャン間隔	Secure Workload が Azure からインベントリデータの増分変更情報を取得する頻度。最小値は 600 秒で、これがデフォルトです。

ステップ 9 [次へ（Next）] をクリックします。システムが Azure から VNet と AKS クラスタのリストを取得するには、数分かかる場合があります。

ステップ 10 VNet と、各 VNet の AKS クラスタのリストから、選択した機能を有効にする VNet と AKS クラスタを選択します。

一般に、Secure Workload が適正なポリシーを提案するのに十分なデータの収集を開始できるように、できるだけ早くフローの取り込みを有効にする必要があります。

AKS はラベルの収集機能のみをサポートしているため、明示的な機能の選択はできないことに注意してください。AKS クラスタを選択すると、サポートされている機能が暗黙的に有効になります。この機能を有効にする各クラスタのクライアント証明書とキーをアップロードします。

通常は、初期設定時に [セグメンテーションの有効化（Enable Segmentation）] を選択しないでください。後で、特定の VNet にセグメンテーションポリシーを適用する準備ができたら、コネクタを編集して、それらの VNet のセグメンテーションを有効にすることができます。Azure Inventory にセグメンテーションポリシーを適用するときのベストプラクティス（316 ページ）を参照してください。

ステップ 11 選択が完了したら、[作成（Create）] をクリックし、検証チェックが完了するまで数分待ちます。

[グループの表示（View Groups）] ページには、前のページで機能を有効にしたすべての VNet が地域別にグループ化されて表示されます。各地域、および各地域の各 VNet は、新しい範囲です。

ステップ 12 （オプション）新しい範囲のセットを追加する親範囲を選択します。いずれの範囲もまだ定義していない場合、唯一のオプションはデフォルトの範囲になります。

ステップ 13 （オプション）ウィザードで行われたすべての設定（階層型範囲ツリーを含む）を受け入れるには、[保存（Save）] をクリックします。

階層型範囲ツリーを除くすべての設定を受け入れるには、[この手順をスキップ（Skip this step）] をクリックします。

範囲ツリーは、後で **[整理（Organize）]** > **[範囲とインベントリ（Scopes and Inventory）]** で手動で作成または編集できます。

次のタスク

ラベルの収集、フローデータの取り込み、および/またはセグメンテーションを有効にしている場合：

- フローの取り込みを有効にした場合、フローが **[調査（Investigate）]** > **[トラフィック（Traffic）]** ページに表示されるまでに最大 25 分かかる場合があります。
- （オプション）より豊富なフローデータと、ホストの脆弱性の可視化（CVE）といったその他の利点を得るため、VNet ベースのワークロードにオペレーティングシステムに適切なエージェントをインストールします。要件と詳細については、エージェントのインストールの章を参照してください。
- ラベルを収集してフローを取り込むように Azure コネクタを正常に設定した後は、セグメンテーションポリシーを構築するための標準プロセスに従います。例：Secure Workload が以下を実行できるようにします。十分なフローデータを収集して信頼できるポリシーを生成する。スコープを定義または変更する（通常は VNet ごとに 1 つ）。範囲ごとにワークスペースを作成する。フローデータに基づいてポリシーを自動的に検出する、および/または手動でポリシーを作成する。ポリシーを分析して改善する。ポリシーが下記のガイドラインとベストプラクティスを満たしていることを確認する。準備ができたら、ワークスペースで該当するポリシーを承認して適用します。特定の VNet にセグメンテーションポリシーを適用する準備ができたら、コネクタ設定に戻って VNet のセグメンテーションを有効にします。詳細については、Azure Inventory にセグメンテーションポリシーを適用するときのベストプラクティス （316 ページ）を参照してください。

Kubernetes マネージドサービス（AKS）オプションを有効にしている場合：

- コンテナベースのワークロードに Kubernetes エージェントをインストールします。詳細については、エージェント展開の章の「優れた可視性と適用のための Kubernetes または OpenShift エージェントのインストール （46 ページ）」を参照してください。

イベントログ：

イベントログを使用すると、さまざまな機能からコネクタごとに発生する重要なイベントを把握することができます。コンポーネント、名前空間、メッセージ、およびタイムスタンプなどのさまざまな属性を使用してフィルタできます。

AWS コネクタの編集

Azure コネクタを編集して、特定の VNet のセグメンテーションの適用を有効にしたり、その他の変更を加えたりできます。

ウィザードを終了するまで、変更は保存されません。

手順

ステップ 1 ウィンドウの左側にあるナビゲーションバーから、**[管理（Manage）]>[コネクタ（Connectors）]** を選択します。

ステップ 2 [Azure] をクリックします。

ステップ 3 Azure コネクタが複数ある場合は、編集するコネクタをウィンドウの上部から選択します。

ステップ 4 [コネクタの編集（Edit Connector）] をクリックします。

ステップ 5 ウィザードをもう一度クリックして、変更を実行します。設定の詳細については、「Azure コネクタの設定 （311 ページ）」を参照してください。

ステップ 6 さまざまな機能（ラベルの収集、フローの取り込み、セグメンテーションの適用、AKS データの収集）を有効にする場合は、改訂版の ARM テンプレートをダウンロードし、新しいテンプレートテキストを編集してサブスクリプション ID を指定する必要があります。また、ウィザードを続行する前に、Azure で作成したカスタムロールに新しいテンプレートをアップロードします。

ステップ 7 セグメンテーションポリシーの適用を有効にするには、「Azure Inventory にセグメンテーションポリシーを適用するときのベストプラクティス （316 ページ）」で推奨されている前提条件を満たしていることを最初に確認してください。次に、VNet が一覧表示されているウィザードページで、適用を有効にする VNet の [セグメンテーションの有効化（Enable Segmentation）] を選択します。

ステップ 8 選択した VNet のいずれかの範囲をすでに作成している場合は、ウィザードを使用するか手動で、[この手順をスキップ（Skip this step）] をクリックしてウィザードを終了します。

[整理（Organize）] > [範囲とインベントリ（Scopes and Inventory）] ページを使用すると、範囲ツリーを手動で編集できます。

ステップ 9 選択した VNet の範囲をまだ作成しておらず、提案された階層を保持する場合は、範囲ツリーの上から親範囲を選択し、[保存（Save）] をクリックします。

コネクタとデータの削除

コネクタを削除しても、そのコネクタによってすでに取り込まれたデータは削除されません。

ラベルとインベントリは、24 時間後にアクティブなインベントリから自動的に削除されます。

Azure Inventory にセグメンテーションポリシーを適用するときのベストプラクティス

警告 VNetでセグメンテーションの適用を有効にする前に、そのVNetでネットワークセキュリティグループのバックアップを作成します。VNet のセグメンテーションを有効にすると、その仮想ネットワークに関連付けられているネットワーク セキュリティ グループから既存のルールが削除されます。セグメンテーションを無効にしても、古いネットワークセキュリティグループは復元されません。

ポリシーの作成時：検出されたすべてのポリシーと同様に、正確なポリシーを生成するのに十分なフローデータがあることを確認してください。

関連付けられている VNet のセグメンテーションを有効にする前に、ワークスペースで適用を有効にすることを推奨します。適用が有効になっているワークスペースに含まれていないVNetのセグメンテーションを有効にすると、その VNet ですべてのトラフィックが許可されます。

VNet にポリシーを適用する準備ができたら、Azure コネクタを編集し（「AWS コネクタの編集 （315 ページ）」を参照）、その VNet のセグメンテーションを有効にします。

サブネットにネットワークセキュリティグループが関連付けられていない場合、Secure Workloadはそのサブネットにセグメンテーションポリシーを適用しないことに注意してください。VNetにセグメンテーションポリシーを適用すると、サブネットレベルのNSGがすべてのトラフィックを許可するように変更され、Secure Workload ポリシーによってインターフェイスレベルのNSG が上書きされます。インターフェイスの NSG がまだ存在しない場合は、自動的に作成されます。

Azure インベントリラベル、詳細、および適用ステータスの表示

Azure コネクタの概要情報を表示するには、コネクタページ（[管理（Manage）] > [コネクタ（Connectors）]）に移動し、ページの上部からコネクタを選択します。詳細については、[VNet] 行をクリックしてください。

Azure VNet インベントリに関する情報を表示するには、[Azureコネクタ（Azure Connectors）] ページで IP アドレスをクリックして、そのワークロードの [インベントリプロファイル（Inventory Profile）]ページを表示します。インベントリプロファイルの詳細については、「インベントリプロファイル」を参照してください。

ラベルの詳細については、次を参照してください。

- クラウドコネクタによって生成されたラベル
- Kubernetes クラスタに関連するラベル

VNetインベントリの具体的なポリシーは、そのOrchestrator_system/interface_id ラベル値に基づいて生成されます。これは、[インベントリプロファイル（Inventory Profile）] ページで確認できます。

適用ステータスを表示するには、Secure Workload ウィンドウの左側のナビゲーションバーから **[保護（Defend）]** > **[適用ステータス（Enforcement Status）]** を選択します。詳細については、「Cloud Connector の適用ステータス」を参照してください。

Azure コネクタに関する問題のトラブルシューティング

問題：Azure が予期せずすべてのトラフィックを許可する

解決策：Secure Workload の Catch-All ポリシーが [拒否（Deny）] に設定されていることを確認します。

Azure（AKS）で実行されるマネージド Kubernetes サービス

Azure クラウドに Azure Kubernetes Services（AKS）を展開した場合は、Azure コネクタを使用して、Kubernetes クラスタからインベントリとラベル（AKS タグ）を動的にプルできます。

マネージド Kubernetes サービスからメタデータをプルするように Azure コネクタが設定されている場合、Secure Workload はそのクラスタ内のノード、ポッド、およびサービスの状態を追跡します。

このコネクタを使用して収集および生成された Kubernetes ラベルについては、「Kubernetes クラスタに関連するラベル」を参照してください。

AKS の要件および前提条件

- 使用している Kubernetes バージョンがサポートされていることを確認します。Cisco Secure Workload エージェントのオペレーティングシステム、外部システム、およびコネクタについては、「互換性マトリックス」を参照してください。
- Azure コネクタを構成するときに、マネージド Kubernetes サービス（AKS）機能を有効にして構成します。詳細については、「Azure コネクタの設定」を参照してください。

AKS ロードバランサのサポート

AKS は、クライアント IP の保持をサポートしています。

次のポリシーインテントの場合：

次のように生成された、さまざまなケースの許可アクションルールを持つ、コンシューマからプロバイダーへのサービス、サービスプロトコル、およびポート：

ケース	クライアントの保持
1	オン
2	オフ

ケース 1：クライアント IP の保持が**オン**になっている。

コンシューマノードで、コンシューマからロードバランササービス（lb 入力 ip）へのサービスプロトコルとポートの許可を使用して出力ルールを生成します。

プロバイダーノード用に事前ルーティングルールが生成されます。このルールは、コンシューマを送信元として指定し、すべてのプロバイダーノードを宛先として指定します。このルールでは、サービスプロトコルをプロトコルとして含め、サービスのノードポートをポートとして含め、アクションを許可に設定します。

プロバイダーポッドでは、送信元をプロバイダーノード、宛先をプロバイダーポッド（任意）、プロトコルをターゲットプロトコル、ポートをターゲットポート、アクションを許可として、入力ルールを生成します。

ケース 2：クライアント IP の保持が**オフ**になっている。

コンシューマノードで、コンシューマからロードバランササービス（lb 入力 ip）へのサービスプロトコルとポートの許可を使用して出力ルールを生成します。

プロバイダーノードは、lb 入力 IP を送信元として設定し、すべてのプロバイダーノードを宛先として設定する、事前ルーティングルールを生成します。このルールでは、サービスプロトコルをプロトコルとして指定し、サービスのノードポートをポートとして指定し、アクションを許可に設定します。

プロバイダーポッドでは、送信元をプロバイダーノード、宛先をプロバイダーポッド（任意）、プロトコルをターゲットプロトコル、ポートをターゲットポート、アクションを許可として、入力ルールを生成します。

GCP コネクタ

Google Cloud Platform コネクタは、GCP に接続して、次の高度な機能を実行します。

- **GCP 仮想プライベートクラウド（VPC）からのインベントリ（およびそのタグ）のライブ自動取り込み**

  GCP では、タグの形式でリソースにメタデータを割り当てることができます。Cisco Secure Workload は、これらのリソースのタグをクエリし、インベントリおよびトラフィックのフローデータの可視化と、ポリシー定義に使用することができます。この機能では、このデータを常に同期することにより、リソースタグのマッピングが最新の状態に保たれます。

  タグは、GCP VPC のワークロードとネットワークインターフェイスから取り込まれます。ワークロードとネットワークインターフェイスの両方が構成されている場合、タグはマージされて Cisco Secure Workload に表示されます。詳細については、クラウドコネクタによって生成されたラベル （410 ページ）を参照してください。

- **VPC からのフローログの取り込み** 監視目的で GCP で VPC フローログを設定している場合、Secure Workload は対応する Google Storage バケットを読み取ることでフローログ情報を取り込むことができます。このテレメトリは、視覚化およびセグメンテーションポリシーの生成に使用できます。

- **セグメンテーション** このオプションを有効にすると、Secure Workload は、GCP のネイティブ VPC ファイアウォールを使用してセキュリティポリシーをプログラムできるようになります。VPC に対して適用が有効になっている場合、関連するポリシーが VPC ファイアウォールに対して自動的にプログラムされます。

- **GKE クラスタからのメタデータの自動取り込み** GCP で Google Kubernetes Engine（GKE）が実行されている場合、K8s 機能により、選択したすべての Kubernetes クラスタに関連するすべてのノード、サービス、およびポッドのメタデータを収集することを選択できます。

上記の機能の中で VPC ごとに有効にする機能を選択できます。

GCP の要件および前提条件

すべての機能：GCP で専用のサービスアカウントを作成するか、このコネクタ用の既存の GCP サービスアカウントを特定します。コネクタ構成ウィザードは、このサービスアカウントに必要な権限を割り当てるために使用できる IAM ポリシーリストを生成します。この IAM ポリシーリストをアップロードするためのアクセス許可が GCP にあることを確認してください。



（注） IAM ポリシーリストのアクセス許可をサービスアカウントに適用するために推奨される方法は、CLI を使用する方法です。

各 VPC は、1 つの GCP コネクタにのみ属することができます。1 つの Secure Workload クラスタは、複数の GCP コネクタを持つことができます。次のGCP コネクタの設定 （322 ページ）の表に記載されている情報を収集します。

このコネクタには、仮想アプライアンスは必要ありません。

ラベルとインベントリを収集するには：追加の前提条件は必要ありません。

フローログを取り込むには：フローログの収集をトリガーするには、VPC レベルのフローログ定義が必要です。

フローログの取り込みを使用するには、目的の VPC でフローログを有効にし、ログルータシンクを設定する必要があります。

ログルータシンクの包含フィルタ：

1. `resource.type="gce-subnetwork"`
2. `log_name="projects/<project_id>/logs/compute.googleapis.com%2Fvpc_flows"`

シンクの宛先をクラウドストレージバケットとして選択してから、目的のストレージバケットを選択してください。

入力フローログを使用して GCP コネクタを設定する場合は、入力としてストレージバケット名を選択して指定する必要があります。

VPC からのフローログのみを取り込むことができます。

フローログは、Google ストレージバケットに公開する必要があります。Secure Workload は、Google Cloud の運用スイートからフローデータを収集できません。

コネクタの作成中に提供された GCP ユーザーアカウントで VPC フローログと Google ストレージバケットの両方にアクセスできる場合、Cisco Secure Workload は、任意のアカウントに関連付けられた Google ストレージバケットからフローログを取り込むことができます。

フローログには、次のフローログ属性（順序は任意）が必要です。送信元アドレス、宛先アドレス、送信元ポート、宛先ポート、プロトコル、パケット数、バイト数、開始時刻、終了時刻、アクション、TCP フラグ、Interface-ID、ログのステータス、フローの方向。その他の属性は無視されます。

フローログは、許可されたトラフィックと拒否されたトラフィックの両方をキャプチャする必要があります。

セグメンテーション：セグメンテーションを有効にするには、ラベルの収集を有効にする必要があります。

VPC のセグメンテーションポリシーの適用を有効にすると、既存のすべてのルールが上書きされるため、コネクタでセグメンテーションを有効にする前に、既存のセキュリティグループをバックアップしてください。

以下のGCP インベントリにセグメンテーションポリシーを適用するときのベストプラクティス（325 ページ）も参照してください。

マネージド Kubernetes サービス（GKE）：Kubernetes オプションを有効にする場合は、必要なアクセス権限を含む、以下の「GCP（GKE）で実行されるマネージド Kubernetes サービス」の項で要件と前提条件を参照してください。

GCP での複数のプロジェクトのアクセスの設定

GCP で複数のプロジェクトのアクセスを設定するには、次の手順に従います。

手順

ステップ 1 GCP コンソールにサインインします。

ステップ 2 上部ナビゲーションバーのプロジェクトドロップダウンメニューをクリックして [新しいプロジェクト（New Project）] を選択するか、サービスアカウントを使用して新しいプロジェクトを作成するか、既存のプロジェクトを使用することができます。

ステップ 3 新しいプロジェクトの名前を入力します。新しいプロジェクトを所有する組織を選択するか、組織がない場合は [組織なし（No organization）] を選択します。

ステップ 4 [作成（Create）] ボタンをクリックして、新しいプロジェクトを作成します。

（注） これらの手順を繰り返して、必要な数のプロジェクトを作成できます。

ステップ 5 単一サービスアカウント内の複数のプロジェクトをリンクします。これを行うには、GCP コンソールの [IAMと管理（IAM & Admin）] ページに移動し、[サービスアカウント（Service Account）] を選択します。

ステップ 6 [サービスアカウントの作成（Create Service Account）] ボタンをクリックします。プロンプトに従ってサービスアカウントを作成し、必要な権限を付与します。

（注） 既存のサービスアカウントを使用するか、新しいサービスアカウントを作成することができます。

ステップ 7 [キー（key）] タブから [キーの追加（add key）] をクリックして、JSON ファイルに秘密キーを生成します。

ステップ 8 GCP コンソールの [IAMと管理（IAM & Admin）] ページに移動し、[IAM] を選択します。

（注） 最初にプロジェクトを変更してから、[IAMと管理（IAM & Admin）] をクリックして権限の付与を試みる必要があります。

ステップ 9 [アクセス権の付与（Grant access）] ボタンをクリックして、新しいプロジェクトを追加します。

ステップ 10 [新規プリンシパル（New principals）] フィールドに、プロジェクトにリンクするサービスアカウントの電子メールアドレスを入力します。

ステップ 11 [保存（Save）] ボタンをクリックして、サービスアカウントをプロジェクトに関連付けます。

（注） 元のプロジェクトにリンクするプロジェクトごとに、これらの手順を繰り返します。

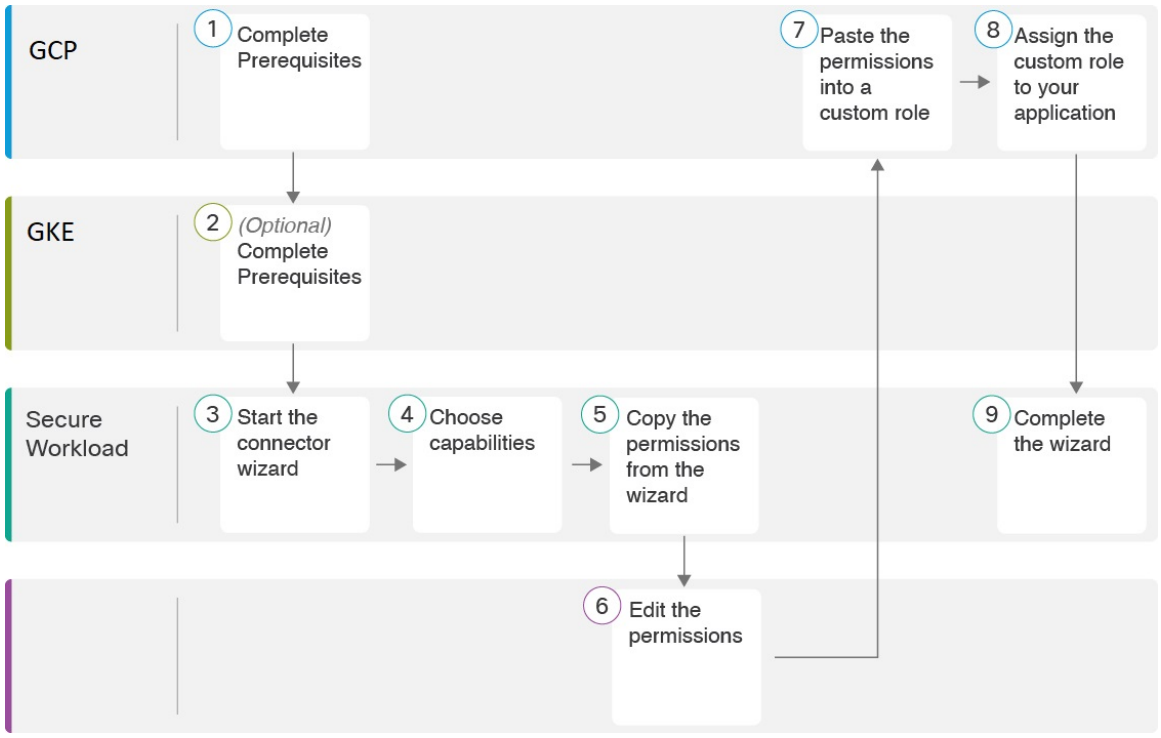
サービスアカウントの権限を管理するには、GCP コンソールの [IAMと管理（IAM & Admin）] ページに移動し、各プロジェクトの [IAM] を選択します。

ステップ 12 サービスアカウントに、最小共通先祖（選択したすべてのプロジェクトに共通の先祖）のリソースレベル（フォルダや組織など）に対する権限があることを確認します。

GCP コネクタ設定の概要

次の図は、コネクタ設定プロセスの高度な概要を示しています。重要な詳細については、次のトピック（GCP コネクタの設定 （322 ページ））を参照してください。

図 *97 : GCP* コネクタ設定の概要



（図中の番号は、詳細手順のステップ番号に対応していないことに注意してください。）

GCP コネクタの設定

手順

ステップ 1 ウィンドウの左側にあるナビゲーションバーから、**[管理（Manage）]**>**[コネクタ（Connectors）]** を選択します。

ステップ 2 該当する GCPコネクタをクリックします。

ステップ 3 （ルート範囲内の）最初のコネクタの場合は[有効化（Enable）] をクリックし、同じルート範囲内の付加的なコネクタの場合は [別のコネクタを有効化（Enable Another）] をクリックします。

ステップ 4 GCP の要件および前提条件 （319ページ）の要件と前提条件を理解し満たしてから、[開始する（Get Started）] をクリックします。

ステップ 5 コネクタに名前を付け、必要な機能を選択して、[次へ（Next）] をクリックします。

このページでの選択は、次の手順で生成される IAM ポリシーリストに含まれる権限を決定し、設定が必要な項目を表示するためにのみ使用されます。

セグメンテーションを有効にするには、[ラベルの収集（Gather Labels）] も有効にする必要があります。

ステップ 6 生成された IAM ポリシーリストをダウンロードします。

この IAM ポリシーリストには、前の手順で選択した機能に必要な IAM 権限があります。

Kubernetes オプションを有効にした場合は、GKE へのアクセス許可を別個に設定する必要があります。下の「GCP（GKE）で実行されるマネージド Kubernetes サービス」の項を参照してください。

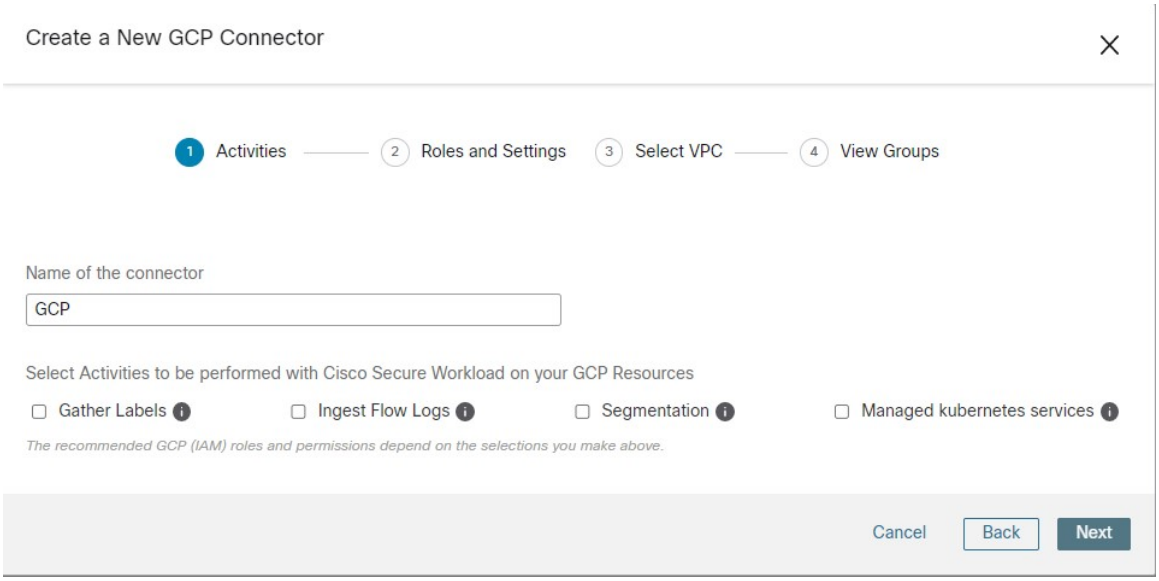
ステップ 7 上記の前提条件で準備した必要な機能を使用して、サービスアカウント json ファイルをアップロードします。

（注） GCP では、単一のコネクタが複数のプロジェクトをサポートするため、サービスアカウントがすべてのプロジェクトに直接リンクされていることを確認してください。

ステップ 8 設定を次のように構成します。

属性	説明
HTTP プロキシ	GCP に到達するために Secure Workload に必要なプロキシ。
フルスキャン間隔	Secure Workload が GCP からの完全なインベントリデータを更新する頻度。最小値は 3600 秒で、これがデフォルトです。

属性	説明
差分スキャン間隔	Secure Workload が GCP からインベントリデータの増分変更情報を取得する頻度。最小値は 600 秒で、これがデフォルトです。



ステップ 9 [次へ（Next）] をクリックします。システムが GCP プロジェクトから仮想ネットワークと GKE クラスタのリストを取得するには、数分かかる場合があります。

ステップ 10 VPC（仮想ネットワーク）と GKE クラスタのリストから、リソースとそれぞれの機能を選択します。

一般に、Secure Workload が適正なポリシーを提案するのに必要な十分なデータの収集を開始できるように、できるだけ早くフローの取り込みを有効にする必要があります。

通常は、初期設定時に [セグメンテーションの有効化（Enable Segmentation）] を選択しないでください。後で、特定の VPC にセグメンテーションポリシーを適用する準備ができたら、コネクタを編集して、それらの VPC のセグメンテーションを有効にすることができます。GCP インベントリにセグメンテーションポリシーを適用するときのベストプラクティスを参照してください。

ステップ 11 選択が完了したら、[作成（Create）] をクリックし、検証チェックが完了するまで数分待ちます。

[グループの表示（View Groups）] ページには、前のページで機能を有効にしたすべての VPC が、project_id（GCP）でもある logical_group_id（CSW）によってグループ化されて表示されます。各 logical_group_id と各 logical_group_id の各 VPC は、新しい範囲です。

ステップ 12 新しい範囲のセットを追加する親範囲を選択します。いずれの範囲もまだ定義していない場合、唯一のオプションはデフォルトの範囲になります。

ステップ 13 ウィザードで行われたすべての設定（階層型範囲ツリーを含む）を受け入れるには、[保存（Save）] をクリックします。

階層型範囲ツリーを除くすべての設定を受け入れるには、[この手順をスキップ（Skip this step）] をクリックします。

範囲ツリーは、後で **[整理（Organize）]** > **[範囲とインベントリ（Scopes and Inventory）]** で手動で作成または編集できます。

次のタスク

ラベルの収集、フローデータの取り込み、および/またはセグメンテーションを有効にしている場合：

- フローの取り込みを有効にした場合、フローが **[調査（Investigate）]** > **[トラフィック（Traffic）]** ページに表示されるまでに最大 25 分かかる場合があります。
- （オプション）より豊富なフローデータと、ホストの脆弱性の可視化（CVE）といったその他の利点を得るため、VPC ベースのワークロードにオペレーティングシステムに適切なエージェントをインストールします。要件と詳細については、エージェントのインストールの章を参照してください。
- ラベルを収集してフローを取り込むように GCP コネクタを正常に設定した後は、セグメンテーションポリシーを構築するための標準プロセスに従います。例：Secure Workload が以下を実行できるようにします。十分なフローデータを収集して信頼できるポリシーを生成する。スコープを定義または変更する（通常は VPC ごとに 1 つ）。範囲ごとにワークスペースを作成する。フローデータに基づいてポリシーを自動的に検出する、および/または手動でポリシーを作成する。ポリシーを分析して改善する。ポリシーが下記のガイドラインとベストプラクティスを満たしていることを確認する。準備ができたら、ワークスペースで該当するポリシーを承認して適用します。特定の VPC にセグメンテーションポリシーを適用する準備ができたら、コネクタ設定に戻って VPC のセグメンテーションを有効にします。詳細については、GCP インベントリにセグメンテーションポリシーを適用するときのベストプラクティス （325 ページ）を参照してください。

Kubernetes マネージドサービス（GKE）オプションを有効にしている場合：

- コンテナベースのワークロードに Kubernetes エージェントをインストールします。詳細については、エージェント展開の章の「優れた可視性と適用のための Kubernetes または OpenShift エージェントのインストール」を参照してください。

イベントログ：

イベントログを使用すると、さまざまな機能からコネクタごとに発生する重要なイベントを把握することができます。コンポーネント、名前空間、メッセージ、およびタイムスタンプなどのさまざまな属性を使用してフィルタできます。

GCP コネクタの編集

異なる VPC または GKE クラスタや追加の VPC または GKE クラスタからのデータ収集を有効にする場合、サービスアカウントの json ファイルのアップロードが必要になる場合があります。これは、必要な機能を使い、異なる権限で VPC または GKE を選択する前に実行します。

ウィザードを終了するまで、変更は保存されません。

手順

ステップ 1 ウィンドウの左側にあるナビゲーションバーから、**[管理（Manage）]** > **[ワークロード（Workloads）]** > **[コネクタ（Connectors）]** を選択します。

ステップ 2 [GCP] をクリックします。

ステップ 3 GCP コネクタが複数ある場合は、編集するコネクタをウィンドウの上部から選択します。

ステップ 4 [コネクタの編集（Edit Connector）] をクリックします。

ステップ 5 ウィザードをもう一度クリックして、変更を実行します。設定の詳細については、「GCP コネクタの設定 （322 ページ）」を参照してください。

ステップ 6 さまざまな機能（ラベルの収集、フローの取り込み、セグメンテーションの適用、またはGKEデータの収集）を有効にする場合は、ウィザードを続行する前に、改訂されたIAMテンプレートをダウンロードして GKE にアップロードする必要があります。

ステップ 7 セグメンテーションポリシーの適用を有効にするには、「GCP インベントリにセグメンテーションポリシーを適用するときのベストプラクティス （325 ページ）」で推奨されている前提条件を満たしていることを最初に確認してください。VPCの一覧が表示されているページで、適用を有効にする VPC の [セグメンテーションの有効化（Enable Segmentation）] を選択します。

ステップ 8 ウィザードを使用するかまたは手動で、選択したいずれかの VPC の範囲をすでに作成している場合は、[この手順をスキップ（Skip this step）] をクリックしてウィザードを完了します。

[整理（Organize）] > **[範囲とインベントリ（Scopes and Inventory）]** ページを使用すると、範囲ツリーを手動で編集できます。

ステップ 9 選択した VPC の範囲をまだ作成しておらず、提案された階層を保持する場合は、範囲ツリーの上から親範囲を選択し、[保存（Save）] をクリックします。

GCP のコネクタとデータの削除

コネクタを削除しても、そのコネクタによってすでに取り込まれたデータは削除されません。

ラベルとインベントリは、24時間後にアクティブなインベントリから自動的に削除されます。

GCP インベントリにセグメンテーションポリシーを適用するときのベストプラクティス

警告 VPCでセグメンテーションの適用を有効にする前に、そのVPCでセキュリティグループのバックアップを作成します。VPC のセグメンテーションを有効にすると、その VPC から既存のセキュリティグループが削除されます。セグメンテーションを無効にしても、古いセキュリティグループは復元されません。

ポリシーの作成時：

- 検出されたすべてのポリシーと同様に、正確なポリシーを生成するための十分なフローデータがあることを確認してください。
- GCP では、ファイアウォールポリシーで許可と拒否両方のルールを使用できます。GCP ではルールの数に非常に厳格な制限があるため、許可リストのみを使用することを推奨します。

関連付けられた VPC のセグメンテーションを有効にする前に、ワークスペースで適用を有効にすることを推奨します。適用が有効になっているワークスペースに含まれていない VPC のセグメンテーションを有効にすると、その VPC ですべてのトラフィックが許可されます。

VPC にポリシーを適用する準備ができたら、GCP コネクタを編集し（GCP コネクタの編集（324 ページ）を参照）、その VPC のセグメンテーションを有効にします。

GKE インベントリラベル、詳細、および適用ステータスの表示

GCP コネクタの概要情報を表示するには、コネクタページ（[管理（Manage）] > [コネクタ（Connectors）]）に移動し、ページの上部からコネクタを選択します。詳細については、VPC の行をクリックしてから、クラスタをクリックして参照してください。

イベントリに関する情報を表示するには、[GCPコネクタ（GCP Connectors）] ページで IP アドレスをクリックして、そのワークロードの [インベントリプロファイル（Inventory Profile）] ページを表示します。インベントリプロファイルの詳細については、「インベントリプロファイル」を参照してください。

ラベルの詳細については、次を参照してください。

- クラウドコネクタによって生成されたラベル
- Kubernetes クラスタに関連するラベル

GCP コネクタに関する問題のトラブルシューティング

問題：[適用ステータス（Enforcement Status）] ページに、具体的ポリシーがスキップされたと表示されます。

解決策：これは、ファイアウォールポリシー内のルールの数が、GCP コネクタで設定されている GCP の制限を超えると発生します。

具体的ポリシーが SKIPPED と表示されている場合、新しいセキュリティグループは実装されておらず、GCP に以前から存在していたセキュリティグループが引き続き有効です。

この問題を解決するには、小さなサブネットを持つ複数のポリシーではなく、1 つのポリシーで大きなサブネットを使用するなどして、ポリシーを統合できるかどうかを確認します。

バックグラウンド：

セグメンテーションを有効にすると、VPC ごとに具体的ポリシーが生成されます。これらの具体的なポリシーは、GCP でファイアウォールポリシーを作成するために使用されます。ただし、GCP と Secure Workload ではポリシーのカウントが異なります。ファイアウォールポリシーで Secure Workload ポリシーを GCP ファイアウォールルールに変換する場合、GCP のカウントメカニズムは複雑です。詳細については、「GCP」を参照してください。

問題：GCP が予期せずすべてのトラフィックを許可する

対処方法：Secure Workload のキャッチオールポリシーが [拒否（Deny）] に設定されていることを確認します。

GCP（GKE）で実行されるマネージド Kubernetes サービス

クラウドコネクタを使用して、Google Cloud Platform（GCP）で実行されている Google Kubernetes Engine（GKE）クラスタからメタデータを収集できます。

コネクタは、選択したすべての Kubernetes クラスタに関連するすべてのノード、サービス、およびポッドのメタデータを収集します。

要件および前提条件

Cisco Secure Workload の要件：このコネクタには仮想アプライアンスは不要です。

プラットフォーム要件：

- このコネクタに必要なアクセスを設定するための権限が GCP にあることを確認してください。
- 各 GKE クラスタは、1 つの GCP コネクタにのみ属することができます。
- 次の「*GCP* コネクタの設定」の表に記載されている情報を収集します。

GKE の要件：

- GKE で必要なアクセス権限を設定する必要があります。
- マネージド K8s 機能をサポートするために、サービスアカウントに必要なロールは次のとおりです。
  - Compute Network Viewer は、GCP 内のすべてのネットワークリソースへの読み取り専用アクセスを許可する IAM ロールです。https://cloud.google.com/compute/docs/access/iam#compute.networkViewer
  - Kubernetes Engine Viewer は、GKE クラスタ内のリソース（ノード、ポッド、GKE API オブジェクトなど）への読み取り専用アクセスを提供する GKE クラスタロールです。https://cloud.google.com/iam/docs/understanding-roles#kubernetes-engine-roles

コネクタ用の仮想アプライアンス

ほとんどのコネクタは Secure Workload 仮想アプライアンスに展開されます。OVA テンプレートを使用して VMware vCenter の ESXi ホストに、または QCOW2 イメージを使用して他の KVM ベースのハイパーバイザーに必要な仮想アプライアンスを展開します。仮想アプライアンスを展開する手順については、「仮想アプライアンスの展開」で説明します。

仮想アプライアンスの種類

仮想アプライアンスを必要とする各コネクタは、2 種類の仮想アプライアンスのいずれかに展開できます。

Cisco Secure Workload Ingest

Cisco Secure Workload Ingest アプライアンスは、さまざまなコネクタから Secure Workload にフロー観測をエクスポートできるソフトウェアアプライアンスです。

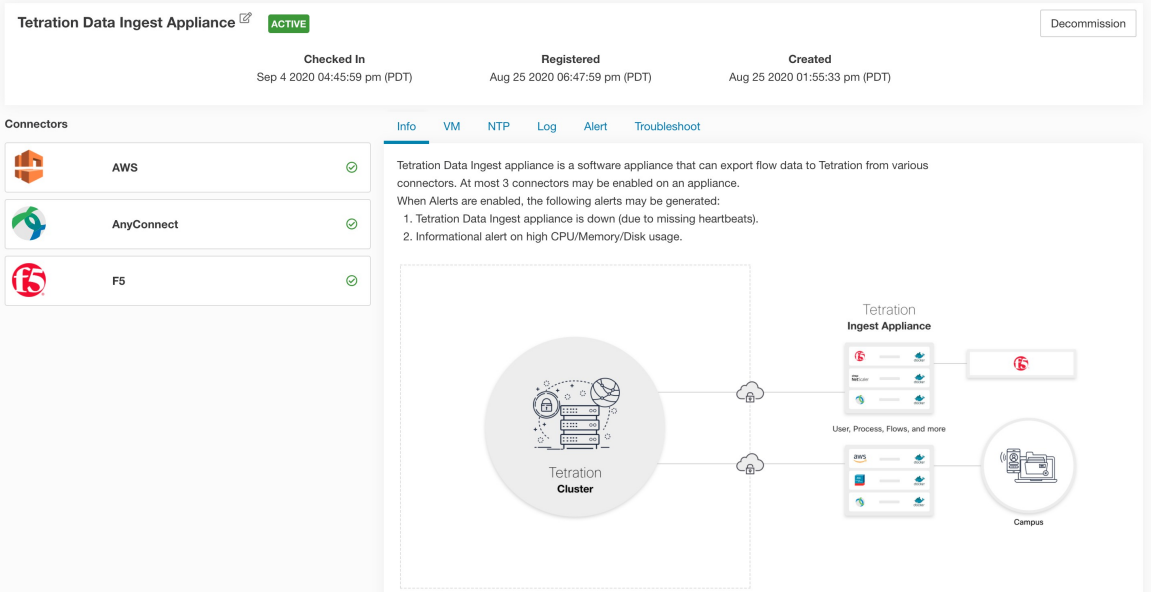
仕様

- CPUコア数：8
- メモリ：8 GB
- ストレージ：250 GB
- ネットワークインターフェイスの数：3
- 1 つのアプライアンスにおけるコネクタ数：3
- オペレーティングシステム：CentOS 7.9

「コネクタ」で、重要な制限事項を確認してください。

（注） Secure Workload のルート範囲には、最大 100 の Secure Workload Ingest アプライアンスを展開できます。

図 *98 : Secure Workload Ingest* アプライアンス



Cisco Secure Workload Ingest アプライアンスでは、1 つのアプライアンスで最大 3 つのコネクタを有効にできます。同じアプライアンスで、同じコネクタの複数のインスタンスを有効にすることができます。ERSPAN Ingest アプライアンスの場合、3 つの ERSPAN コネクタが常に自動的にプロビジョニングされます。Ingest アプライアンスに展開されたコネクタの多くは、ネットワーク内のさまざまなポイントからテレメトリを収集します。これらのコネクタは、アプライアンスの特定のポートでリッスンする必要があります。そのため、各コネクタは、テレメトリデータを収集するためにコネクタがリッスンする必要がある IP アドレスとデフォルトポートのいずれかにバインドされます。その結果、各 IP アドレスは基本的に、アプライアンス上でコネクタが占有するスロットになります。コネクタが有効になると、スロットが使用されます（これにより、スロットに対応する IP が使用されます）。また、コネクタが無効になると、コネクタが占有していたスロットが解放されます（これにより、スロットに対応する IP が解放されます）。Ingest アプライアンスがスロットの状態を維持する方法の説明については、「*Cisco Secure Workload Ingest* アプライアンスのスロット」を参照してください。

図 *99 : Secure Workload Ingest* アプライアンスのスロット


```
[root@beretta-ingest-1 tetter]# cat /local/tetration/appliance/appliance.conf
{
  "type": "TETRATION_DATA_INGEST",
  "slots": [
    {
      "available": false,
      "index": 0,
      "mapped_ip": "172.29.142.26",
      "share_volume": true,
      "count": 1,
      "service_containers": {
        "5d379fac6e37d85f2bdeff45": {
          "connector_id": "5d379fac6e37d85f2bdeff44",
          "service_id": "5d379fac6e37d85f2bdeff45",
          "container_id": "2c7a7ed4f853e85f3d620c663f1c7f5395b53b9dd6696276ac439d34fe142bf1",
          "image_name": "netflow_sensor-3.4.2.52222.maarumug.mrpm.build-netflow:5d379fac6e37d85f2bdeff45",
          "container_name": "nf-5d379fac6e37d85f2bdeff45",
          "service_type": "NETFLOW_SENSOR",
          "ip_bindings": [
            {
              "ip": "172.29.142.26",
              "port": "4729",
              "protocol": "udp"
            },
            {
              "ip": "172.29.142.26",
              "port": "4739",
              "label": 1,
              "protocol": "udp"
            }
          ],
          "volume_id": "373b5b682a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439"
        }
      }
    },
    {
      "available": true,
      "index": 1,
      "mapped_ip": "172.29.142.27",
      "share_volume": true,
      "count": 0,
      "service_containers": null
    },
    {
      "available": true,
      "index": 2,
      "mapped_ip": "172.29.142.28",
      "share_volume": true,
      "count": 0,
      "service_containers": null
    }
  ]
}[root@beretta-ingest-1 tetter]#
```

可能な設定

- NTP：アプライアンスで NTP を設定します。詳細については、「NTP の設定（NTP Configuration）」を参照してください。
- ログ：アプライアンスのログを設定します。詳細については、「ログ設定」を参照してください。

Cisco Secure Workload Edge

Cisco Secure Workload Edge は、さまざまな通知者にアラートをストリーミングし、Cisco ISE などのネットワーク アクセス コントローラからインベントリメタデータを収集する、制御アプライアンスです。Secure Workload Edge アプライアンスでは、すべてのアラート通知コネクタ（Syslog、電子メール、Slack、PagerDuty、Kinesisなど）、ServiceNow コネクタ、ワークロード AD コネクタ、および ISE コネクタを展開できます。

仕様

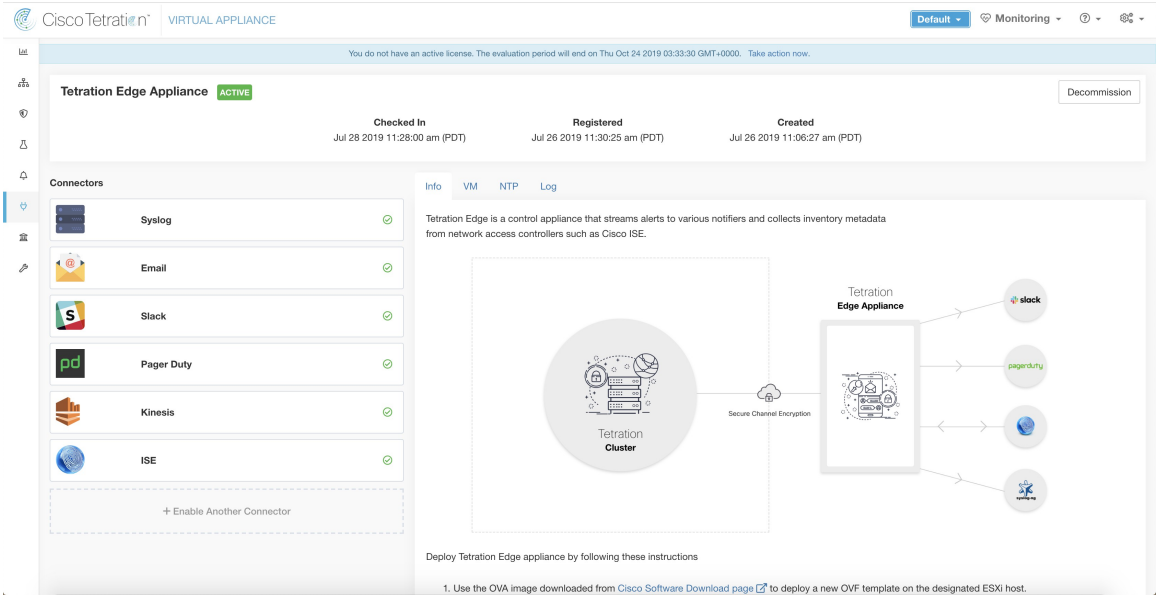
- CPUコア数：8
- メモリ：8 GB
- ストレージ：250 GB
- ネットワークインターフェイスの数：1
- 1 つのアプライアンスのコネクタの数：8
- オペレーティングシステム：CentOS 7.9

「コネクタ」で、重要な制限事項を確認してください。



（注） Secure Workload のルート範囲には、最大 1 つの Secure Workload Edge アプライアンスを展開できます。

図 *100 : Secure Workload Edge* アプライアンス



Secure Workload Edge アプライアンスに展開されたコネクタは、ポートでリッスンしません。したがって、Secure Workload Edge アプライアンスのコネクタ用にインスタンス化された Docker コンテナは、ポートをホストに公開しません。

可能な設定

- NTP：アプライアンスで NTP を設定します。詳細については、「NTP の設定（NTP Configuration）」を参照してください。
- ログ：アプライアンスのログを設定します。詳細については、「ログ設定」を参照してください。

仮想アプライアンスの展開

VMware vCenter または Red Hat Virtualization などの他の KVM ベースのハイパーバイザの ESXi ホストに仮想アプライアンスを展開します。この手順では、シスコ ソフトウェア ダウンロード ページから仮想アプライアンス OVA テンプレートまたは QCOW2 イメージをダウンロードするように求められます。

注目 Secure Workload 外部アプライアンスを展開するには、アプライアンスが作成される ESXi ホストが次の仕様を満たす必要があります。

- **vSphere**：バージョン 5.5 以降。
- **CPU**：コアごとに少なくとも 2.2 GHz であり、アプライアンス用に十分な予約可能な容量があること。
- **メモリ**：少なくともアプライアンスが収まる十分なスペースがあること。

コネクタからデータを収集するために仮想アプライアンスを展開するには、次の手順を実行します。

手順

ステップ 1 Secure Workload Web ポータルで、左側のナビゲーションバーから**[管理（Manage）] > [仮想アプライアンス（Virtual Appliances）]**を選択します。

ステップ 2 [コネクタの有効化（Enable a Connector）] をクリックします。展開する必要のある仮想アプライアンスのタイプは、有効化するコネクタのタイプによって異なります。

ステップ 3 仮想アプライアンスを作成する必要があるコネクタのタイプをクリックします。たとえば、NetFlow コネクタをクリックします。

ステップ 4 コネクタページで、[有効化（Enable）] をクリックします。

ステップ 5 仮想アプライアンスを展開する必要があるという通知が表示されたら、[はい（Yes）] をクリックします。この通知が表示されない場合、このコネクタが使用できる仮想アプライアンスが既にある可能性があります。その場合、この手順を実行する必要はありません。

ステップ 6 リンクをクリックして、仮想アプライアンスの OVA テンプレートまたは QCOW2 イメージをダウンロードします。何もクリックせずに、画面上のウィザードを開いたままにします。

ステップ 7 ダウンロードしたものに応じて、次の手順を実行します。

- OVA：指定された ESXi ホストに新しい OVF テンプレートを展開します。
  - vSphere Web クライアントに OVA を展開する方法については、OVF テンプレートの展開に従ってください。
  - 展開した VM 設定が、仮想アプライアンスのタイプに推奨される設定と一致していることを確認します。
  - **展開した VM の電源をオンにしないでください。**
- QCOW2 イメージ：Red Hat Virtualization などの KVM ハイパーバイザで新しい VM を作成します。

ステップ 8 VM の展開後、電源をオンにする前に、Secure Workload Web ポータルの仮想アプライアンス展開ウィザードに戻ります。

ステップ 9 仮想アプライアンスの展開ウィザードで、[次へ（Next）] をクリックします。

ステップ 10 IP アドレス、ゲートウェイ、ホスト名、DNS、プロキシサーバー設定、および Docker ブリッジサブネット設定を指定して、仮想アプライアンスを設定します。「ネットワークパラメータを使用した *VM* の設定」のスクリーンショットを参照してください。

（注） NetFlow、ERSPAN、および ISE コネクタの場合、IPv6 アドレス（デュアルスタックモード）を指定できます。ただし、デュアルスタックのサポートはベータ版の機能であることに注意してください。デュアルスタックモードの要件と制限の詳細については、Cisco Secure Workload アップグレードガイド [英語] を参照してください。

- アプライアンスがプロキシサーバーを使用して Cisco Secure Workload にアクセスする必要がある場合は、[プロキシサーバーを使用してCisco Secure Workloadに接続（Use proxy server to connect to Secure Workload）] チェックボックスをオンにしてください。これが正しく設定されていない場合、コネクタは、メッセージの制御、コネクタの登録、およびフローデータの Secure Workload コレクタへの送信のために Secure Workload と通信することができない場合があります。
- アプライアンスの IP アドレスとゲートウェイがデフォルトの Docker ブリッジサブネット（172.17.0.1/16）と競合する場合、[Dockerブリッジ（CIDR形式）（Docker Bridge (CIDR format)）] フィールドで指定したカスタマイズした Docker ブリッジサブネットを使用してアプライアンスを設定できます。これには、アプライアンス OVA 3.3.2.16 以降が必要です。

ステップ 11 [次へ（Next）] をクリックします。

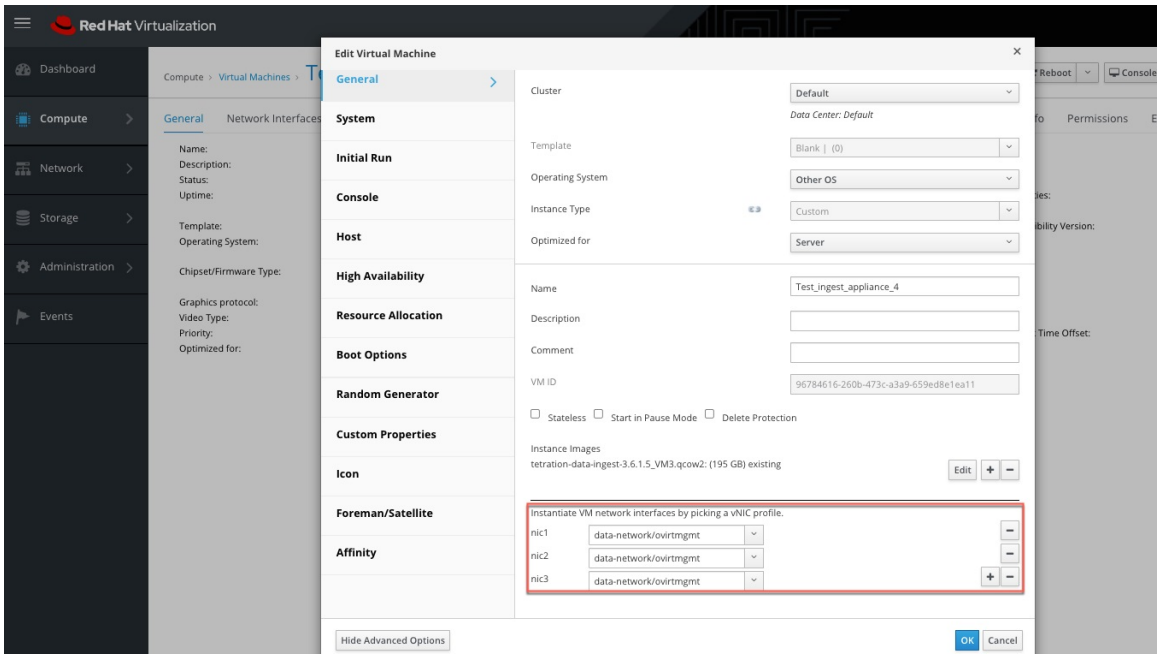
ステップ 12 次のステップで、VM 設定バンドルが生成され、ダウンロードできるようになります。VM 設定バンドルをダウンロードします。「*VM* 設定バンドルのダウンロード」のスクリーンショットを参照してください。

ステップ 13 VM 設定バンドルを、ターゲットの ESXi ホストまたは他の仮想化ホストに対応するデータストアにアップロードします。

ステップ 14 （**QCOW2 イメージを使用する場合のみ適用**）VM 設定バンドルをアップロードした他の仮想化ホストで、次の設定を完了します。

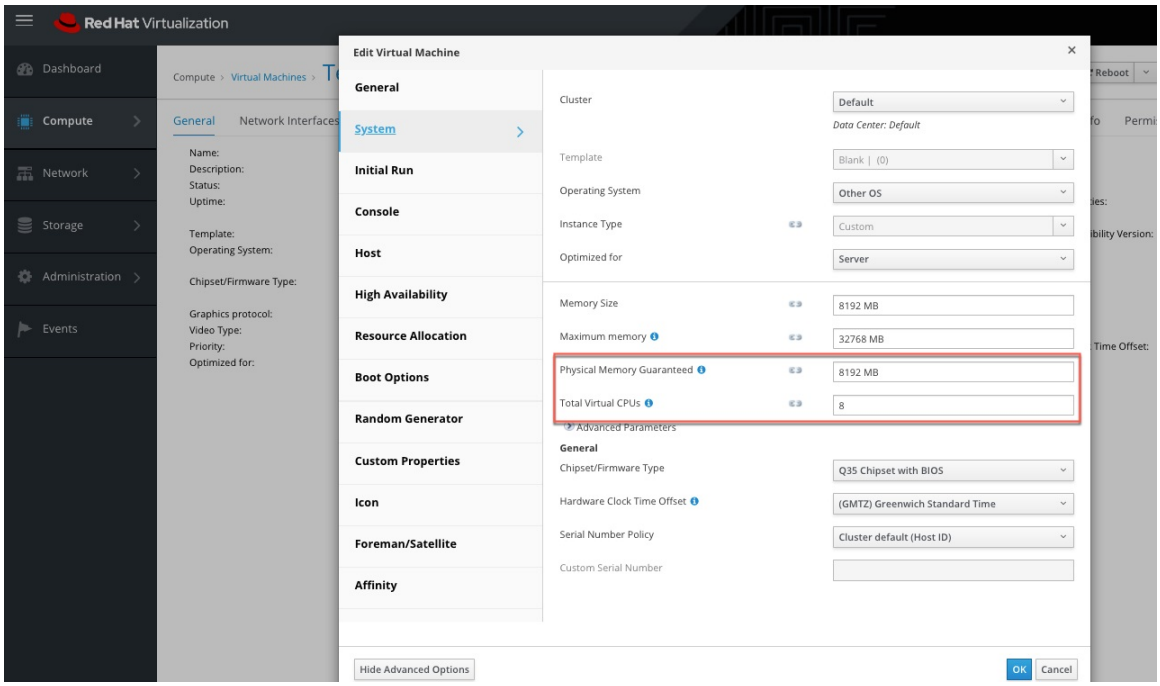
- Ingest アプライアンスの場合、3 つのネットワーク インターフェイスを設定します。

図 *101 : KVM* ベースの環境でのネットワーク インターフェイスの設定例



- メモリ割り当てで、RAM の最小要件を 8192 MB に指定します。
- 仮想 CPU の合計数を 8 に指定します。

図 *102 : KVM* ベースの環境でのシステムリソースの設定例



ステップ 15 VM 設定を編集し、VM 設定バンドルをデータストアから CD/DVD ドライブにマウントします。[電源投入時に接続（Connect at Power On）] チェックボックスをオンにしていることを確認します。

ステップ 16 展開した VM の電源をオンにします。

ステップ 17 VM が起動して自動的に設定を行い、Cisco Secure Workload に接続し直します。数分かかることがあります。Secure Workload 上のアプライアンスのステータスが、[登録の保留中（Pending Registration）] から [アクティブ（Active）] に移行するはずです。「登録保留中の状態の *Cisco Secure Workload Ingest* アプライアンス」のスクリーンショットを参照してください。

（注） Secure Workload 外部アプライアンスに対して vMotion を有効にすることはお勧めしません。

（注） Secure Workload 外部アプライアンス OVA をそのまま使用し、VM を展開するため QCOW2 イメージ用に 8 つの vCPU コアと 8192 MB のメモリを予約することをお勧めします。十分なリソースが利用できない場合、VM セットアップスクリプトは起動後に失敗します。

アプライアンスがアクティブになると、コネクタを有効にして展開することができます。

図 *103 : Secure Workload Ingest* アプライアンスの展開

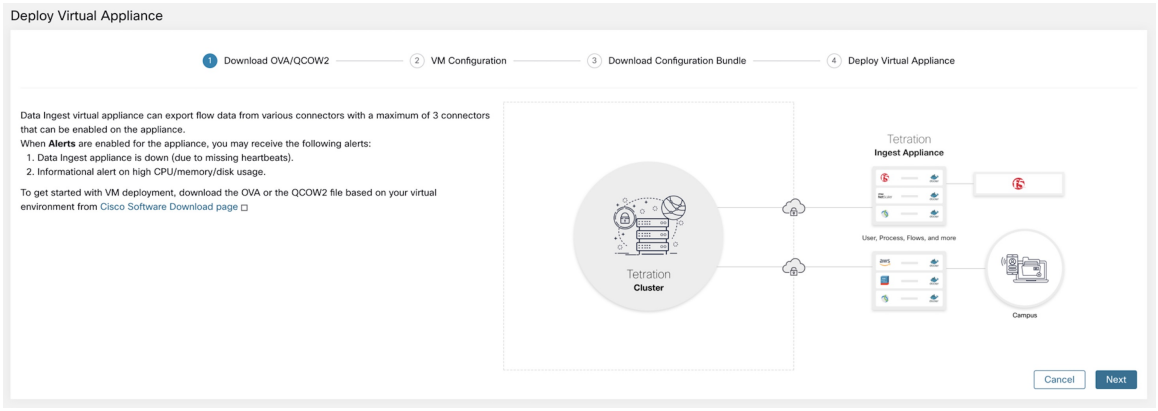


図 *104* : ネットワークパラメータを使用した *VM* の設定

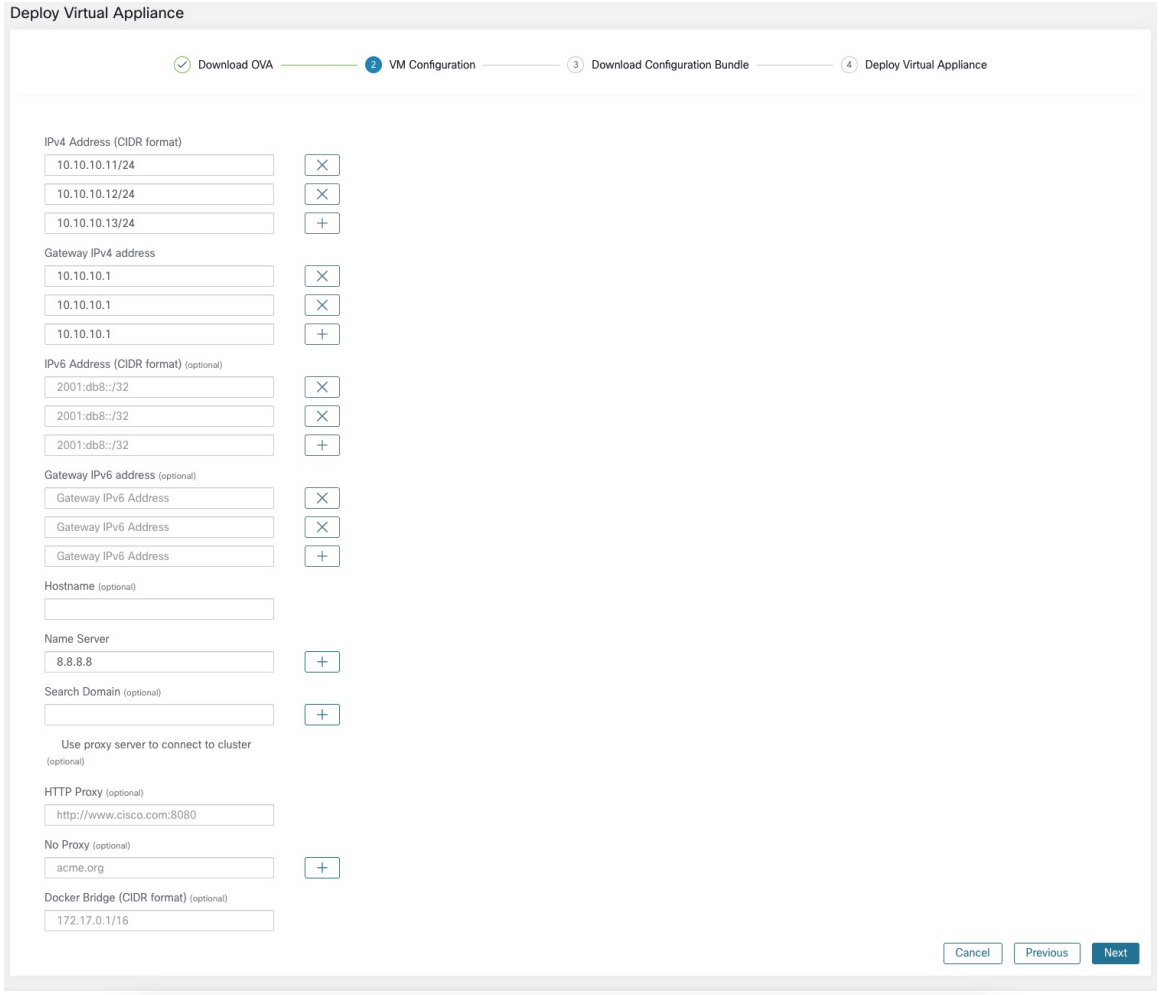


図 *105* : *VM* 設定バンドルのダウンロード

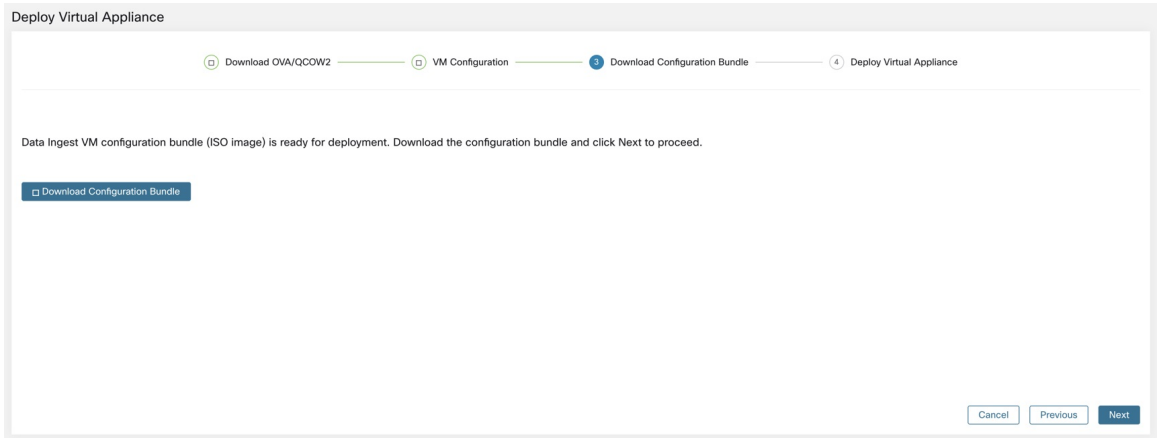


図 *106 : VM* の展開

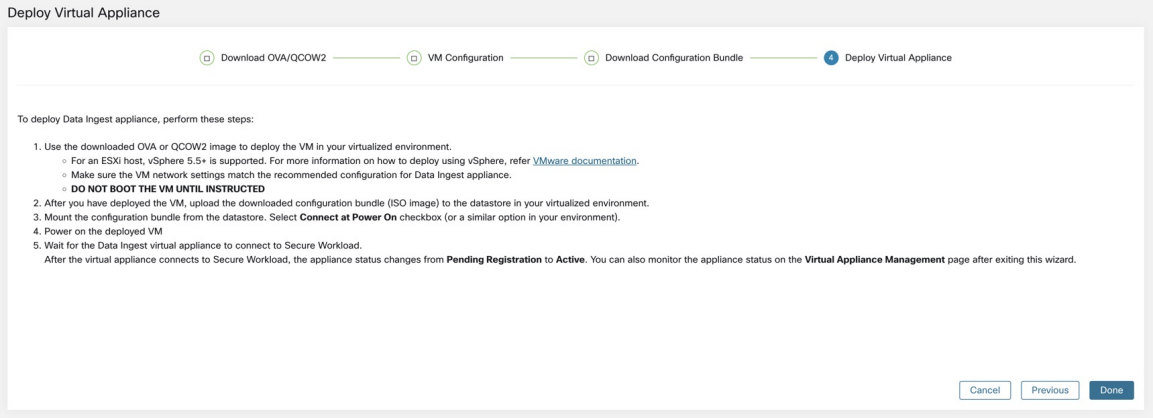
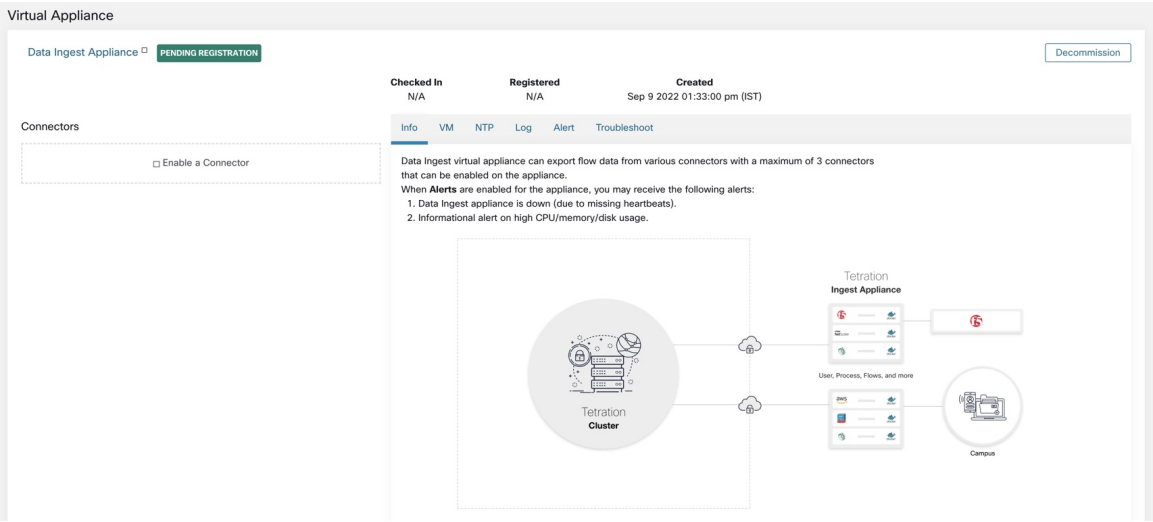


図 *107 :* 登録保留中の状態の *Secure Workload Ingest* アプライアンス



仮想アプライアンスを展開した後に初めて起動すると、*tet-vm-setup* サービスが実行され、アプライアンスがセットアップされます。このサービスは次の役割を果たします。

1. **アプライアンスの検証**：展開された仮想アプライアンスのタイプに必須のリソース要件について、アプライアンスを検証します。
2. **IP アドレスの割り当て**：アプライアンスでプロビジョニングされたすべてのネットワークインターフェイスに IP アドレスを割り当てます。
3. **ホスト名の割り当て**：アプライアンスのホスト名を割り当てます（ホスト名が設定されている場合）。
4. **DNS 設定**：DNS resolv.conf ファイルを更新します（ネームサーバーおよび/または検索ドメインのパラメータが設定されている場合）。
5. **プロキシサーバー設定**：アプライアンスの HTTPS_PROXY および NO_PROXY 設定を更新します（提供されている場合）。

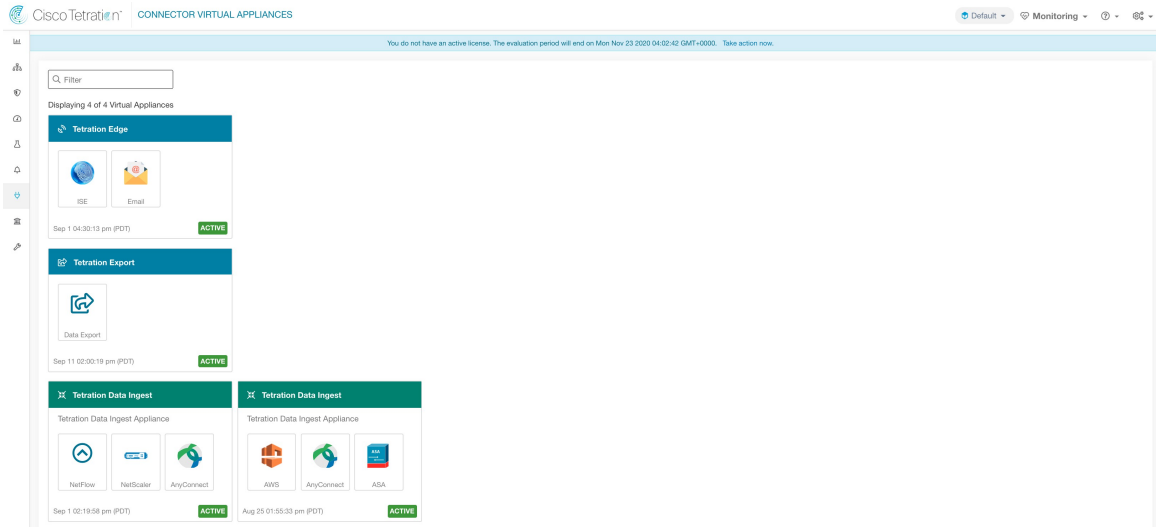
6. **アプライアンスの準備**：アプライアンス管理メッセージが送受信される Kafka トピックの証明書バンドルをコピーします。
7. **アプライアンスコントローラのインストール**：*tet-controller* サービスとして *supervisord* によって管理されるアプライアンスコントローラをインストールして起動します。

tet-controller がインスタンス化されると、アプライアンスの管理を引き継ぎます。このサービスは次の役割を果たします。

1. **登録**：アプライアンスを Cisco Secure Workload に登録します。アプライアンスが登録されるまで、アプライアンスでコネクタを有効にすることはできません。Secure Workload は、アプライアンスの登録要求を受信すると、アプライアンスの状態をアクティブに更新します。
2. **コネクタの展開**：コネクタを Docker サービスとしてアプライアンスに展開します。詳細については、「コネクタの有効化」を参照してください。
3. **コネクタの削除**：Docker サービスおよび対応する Docker イメージを停止して、アプライアンスから削除します。詳細については、「コネクタの削除」を参照してください。
4. **アプライアンスの設定の更新**：アプライアンスの設定の更新をテストして適用します。詳細については、「コネクタおよび仮想アプライアンスの構成管理」を参照してください。
5. **アプライアンスのトラブルシューティング コマンド**：アプライアンスの問題をトラブルシューティングおよびデバッグするために、アプライアンスで許可されたコマンドを実行します。詳細については、「トラブルシューティング」を参照してください。
6. **ハートビート**：定期的にハートビートと統計を Secure Workload に送信して、アプライアンスの状態をレポートします。詳細については、「仮想アプライアンスのモニタリング」を参照してください。
7. **プルーニング**：ストレージスペースを回復するために、未使用またはダングリング状態のすべての Docker リソースを定期的にプルーニングします。このタスクは 24 時間に 1 回実行されます。
8. **アプライアンスのデコミッション**：アプライアンスからすべての Docker インスタンスをデコミッションして削除します。詳細については、「仮想アプライアンスのデコミッション」を参照してください。

展開済みの仮想アプライアンスのリストは、**[管理（Manage）]>[仮想アプライアンス（Virtual Appliances）]**にあります。

図 *108* : 展開済みの仮想アプライアンスのリスト



仮想アプライアンスのデコミッション

仮想アプライアンスは、Cisco Secure Workload からデコミッションできます。アプライアンスがデコミッションされると、次のアクションがトリガーされます。

1. アプライアンスのすべての設定と、アプライアンスで有効になっているコネクタが削除されます。
2. アプライアンスで有効になっているすべてのコネクタが削除されます。
3. アプライアンスには、削除保留のマークが付けられています。
4. アプライアンスが削除成功の応答を返すと、アプライアンスの Kafka トピックと証明書が削除されます。

（注） アプライアンスをデコミッションすると、元に戻すことはできません。アプライアンスとコネクタを復元するには、新しいアプライアンスを展開し、新しいアプライアンスでコネクタを有効にする必要があります。

仮想アプライアンスのモニタリング

Cisco Secure Workload 仮想アプライアンスは、定期的にハートビートと統計を Cisco Secure Workload に送信します。ハートビートの間隔は5分です。ハートビートメッセージには、システム統計やプロセス統計、アプライアンス管理に使用される Kafka トピックを介した送信/受信/エラーメッセージの数に関する統計など、サービスの正常性に関する統計が含まれます。

すべてのメトリックは Digger（OpenTSDB）で使用でき、アプライアンス ID とルート範囲名でラベル付けされます。さらに、アプライアンスコントローラの Grafana ダッシュボードも、アプライアンスからの重要なメトリックに使用できます。

セキュリティに関する注意事項

Ingest/Edge 仮想マシンのゲスト オペレーティング システムは CentOS 7.9 ですが、そこから OpenSSL サーバーおよびクライアントパッケージが削除されています。そのため、アプライアンスにアクセスするにはそのコンソールを使用するしか方法がありません。

コンテナは、centos:7.9.2009 ベースの Docker イメージを実行します。また、NET_ADMIN ケーパビリティを持つ ERSPAN コンテナを除き、ほとんどのコンテナは基本権限（特権なしのオプション）で実行されます。

万が一、コンテナが侵害された場合でも、VM ゲスト OS はコンテナの内部から侵害されないようにするべきです。

コネクタのライフサイクル管理

コネクタは、有効化、展開、構成、トラブルシューティング、および Secure Workload からの直接削除が可能です。

コネクタの有効化

[コネクタ（Connectors）]ページ（**[管理（Manage）]**>**[コネクタ（Connectors）]**）から、コネクタを選択して有効にすることができます。コネクタは、新しい仮想アプライアンス（アプライアンスでコネクタを有効にする前にまずプロビジョニングしてアクティブにする必要があります）または既存の仮想アプライアンスに展開できます。仮想アプライアンスを選択すると、Secure Workload はコネクタの rpm パッケージをアプライアンスに送信します。

選択したアプライアンスのアプライアンスコントローラが rpm を受信すると、次の操作を実行します。

1. Cisco Secure Workload から受け取った rpm パッケージを使用して、Docker イメージを構築します。この Docker イメージには、アプライアンス管理メッセージが送信される Kafka トピックと通信するために必要な設定が含まれています。この設定により、Docker イメージからインスタンス化されたサービスは、対応するコネクタを管理するためのメッセージを送受信できるようになります。
2. Docker イメージから Docker コンテナを作成します。
3. Secure Workload Ingest アプライアンスでは、次の追加タスクが実行されます。
   - 空きスロットが特定され、対応する IP アドレスが決定されます。

- コネクタのリスニングポート（たとえば、NetFlow V9 または IPFIX 対応のスイッチおよびルータからフローレコードを受信する NetFlow コネクタの 4729 および 4739 ポート）が、選択されたスロットに対応する IP 上のホストに公開されます。
- Docker ボリュームが作成され、コンテナに追加されます。

4. Docker コンテナが開始され、コネクタが *supervisord* マネージドサービスとして実行されます。サービスは、Secure Workload に登録されて実際のコネクタサービスを生成する *tet-controller* としてサービスコントローラを開始します。

図 *109 : Docker* イメージ（*Docker Images*）

```
[root@beretta-ingest-1 tetter]# docker images
REPOSITORY                                                TAG                 IMAGE ID            CREATED              SIZE
netflow_sensor-3.4.2.52222.maarumug.mrpm.build-netflow    5d379fac6e37d85f2bdeff45   2635145b44c8        About a minute ago   650MB
tet-service-base                                          latest              6be171bbe648        4 days ago           519MB
artifacts.tet.wtf:6555/centos                             7.3.1611            c5d48e81b986        4 months ago         192MB
[root@beretta-ingest-1 tetter]#
```

図 *110 : Docker* ボリューム

```
[root@beretta-ingest-1 tetter]# docker volume ls
DRIVER              VOLUME NAME
local               373b5b682a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439
[root@beretta-ingest-1 tetter]#
```

図 *111 : Docker* コンテナ

```
[root@beretta-ingest-1 tetter]# docker ps
CONTAINER ID        IMAGE                                                                                COMMAND                  CREATE
D              STATUS              PORTS                                                          NAMES
2c7a7ed4f853        netflow_sensor-3.4.2.52222.maarumug.mrpm.build-netflow:5d379fac6e37d85f2bdeff45   "/usr/bin/supervisor…"   About
a minute ago   Up About a minute   172.29.142.26:4729->4729/udp, 172.29.142.26:4739->4739/udp   nf-5d379fac6e37d85f2bdeff45
[root@beretta-ingest-1 tetter]#
```

図 *112 : Docker* コンテナが使用するスロットと公開されているポートのリスト


```
[root@beretta-ingest-1 tetter]# cat /local/tetration/appliance/appliance.conf
{
  "type": "TETRATION_DATA_INGEST",
  "slots": [
    {
      "available": false,
      "index": 0,
      "mapped_ip": "172.29.142.26",
      "share_volume": true,
      "count": 1,
      "service_containers": {
        "5d379fac6e37d85f2bdeff45": {
          "connector_id": "5d379fac6e37d85f2bdeff44",
          "service_id": "5d379fac6e37d85f2bdeff45",
          "container_id": "2c7a7ed4f853e85f3d620c663f1c7f5395b53b9dd6696276ac439d34fe142bf1",
          "image_name": "netflow_sensor-3.4.2.52222.maarumug.mrpm.build-netflow:5d379fac6e37d85f2bdeff45",
          "container_name": "nf-5d379fac6e37d85f2bdeff45",
          "service_type": "NETFLOW_SENSOR",
          "ip_bindings": [
            {
              "ip": "172.29.142.26",
              "port": "4729",
              "protocol": "udp"
            },
            {
              "ip": "172.29.142.26",
              "port": "4739",
              "label": 1,
              "protocol": "udp"
            }
          ],
          "volume_id": "373b5b682a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439"
        }
      }
    },
    {
      "available": true,
      "index": 1,
      "mapped_ip": "172.29.142.27",
      "share_volume": true,
      "count": 0,
      "service_containers": null
    },
    {
      "available": true,
      "index": 2,
      "mapped_ip": "172.29.142.28",
      "share_volume": true,
      "count": 0,
      "service_containers": null
    }
  ]
}[root@beretta-ingest-1 tetter]#
```


図 *113 : Docker* コンテナによって公開されるポートのリスト


```
[root@beretta-ingest-1 tetter]# docker port 2c7a7ed4f853
4729/udp -> 172.29.142.26:4729
4739/udp -> 172.29.142.26:4739
[root@beretta-ingest-1 tetter]#
```


図 *114 :* コンテナにマウントされた *Docker* ボリューム


```
[root@beretta-ingest-1 tetter]# docker inspect --format='{{json .Mounts}}' 2c7a7ed4f853
[{"Type":"volume","Name":"373b5b682a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439","Source":"/var/lib/docker/volumes/373b5b6
82a96547bf2526784a5943c2f110593b88485b996e7259fa4e314c439/_data","Destination":"/local/tetration","Driver":"local","Mode":"z","RW":tr
ue,"Propagation":""}]
[root@beretta-ingest-1 tetter]#
```


サービスコントローラは、次の役割を果たします。

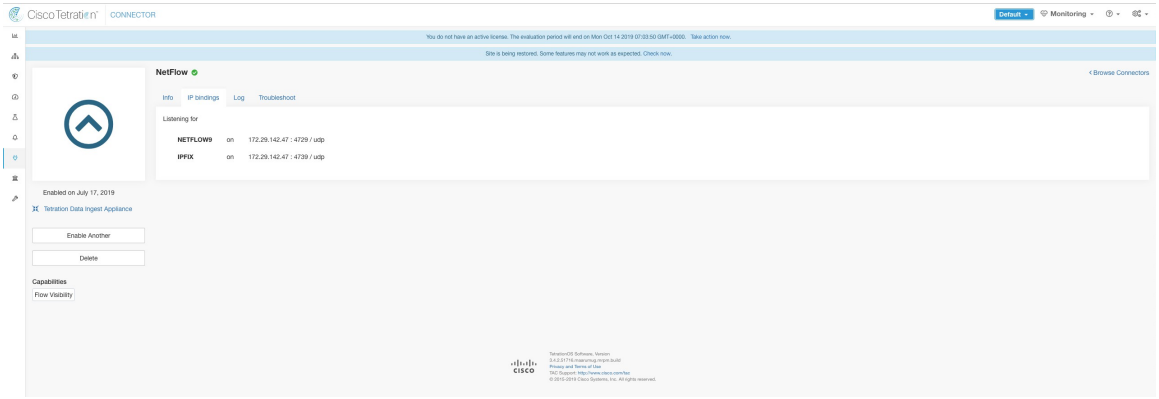
1. **登録**：コネクタを Cisco Secure Workload に登録します。コネクタが登録されて有効と表示されるまで、設定の更新をコネクタにプッシュすることはできません。Secure Workload は、コネクタの登録要求を受信すると、コネクタの状態を有効に更新します。
2. **コネクタの設定の更新**：コネクタの設定の更新をテストして適用します。詳細については、「コネクタおよび仮想アプライアンスの構成管理」を参照してください。
3. **コネクタのトラブルシューティングコマンド**：コネクタサービスの問題をトラブルシューティングおよびデバッグするために、コネクタサービスで許可されたコマンドを実行します。詳細については、「トラブルシューティング」を参照してください。
4. **ハートビート**：定期的にハートビートと統計を Secure Workload に送信して、コネクタの状態をレポートします。詳細については、「仮想アプライアンスのモニタリング」を参照してください。

コネクタ関連情報の表示

有効なコネクタ：ウィンドウ左側にあるナビゲーションバーの **[管理（Manage）]** > **[コネクタ（Connectors）]** をクリックすると、有効なすべてのコネクタのリストが表示されます。

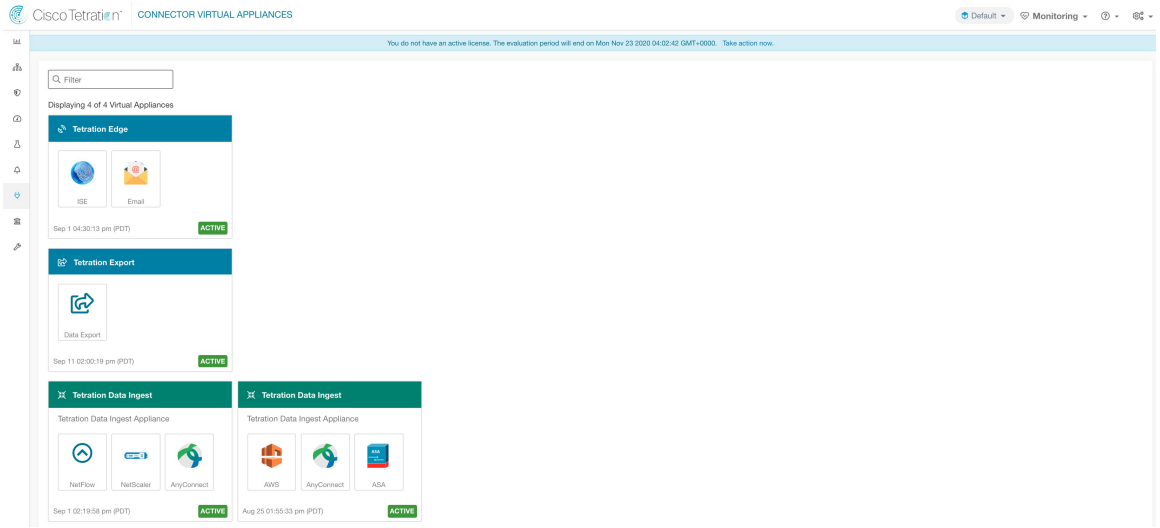
コネクタの詳細：コネクタをクリックすると、コネクタの詳細情報を取得できます。このページには、ポートバインディングが（存在する場合）表示されます。ポートバインディングは、テレメトリデータを正しい IP とポートに送信するようにアップストリーム ネットワーク要素を設定するために使用できます。

図 *115*：コネクタの詳細



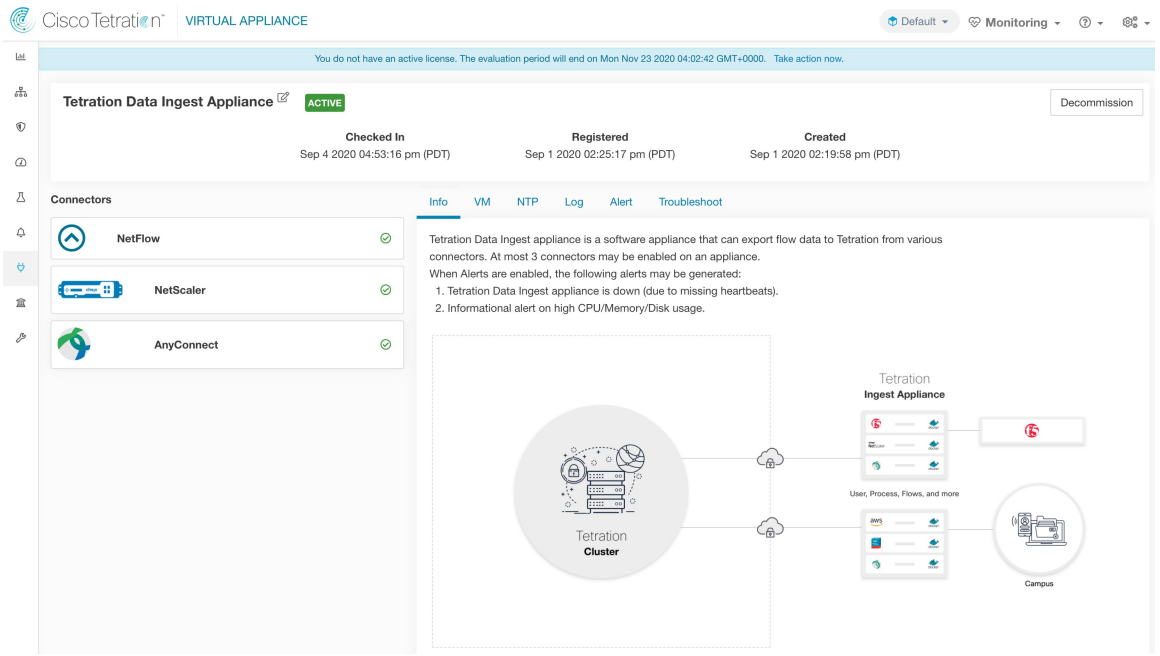
展開済みの仮想アプライアンス：展開済みの仮想アプライアンスのリストは、**[管理（Manage）]** > **[仮想アプライアンス（Virtual Appliances）]** にあります。

図 *116* : 展開済みの仮想アプライアンスのリスト



仮想アプライアンスの詳細：展開済みの仮想アプライアンスのリストでアプライアンスを直接クリックすると、クリックしたアプライアンスの詳細ビューを取得できます。

図 *117* : アプライアンスの詳細とコネクタ



コネクタの削除

コネクタが削除されると、そのコネクタが有効になっているアプライアンスのコントローラは、コネクタ用に作成されたサービスを削除するメッセージを受け取ります。アプライアンスコントローラは次のことを行います。

1. コネクタに対応する Docker コンテナを停止します。
2. Docker コンテナを削除します。
3. コネクタが Secure Workload Ingest アプライアンスに展開され、ポートを公開している場合は、コンテナにマウントされた Docker ボリュームを削除します。
4. コネクタ用に作成された Docker イメージを削除します。
5. 最後に、削除リクエストのステータスを示すメッセージを Secure Workload に送信します。

コネクタのモニタリング

コネクタサービスは、定期的にハートビートと統計を Cisco Secure Workload に送信します。ハートビートの間隔は 5 分です。ハートビートメッセージには、システム統計、プロセス統計、およびアプライアンス管理に使用される Kafka トピックを介して送信/受信/エラーが発生したメッセージの数に関する統計など、サービスの正常性に関する統計が含まれます。さらに、コネクタサービス自体によってエクスポートされた統計も含まれます。

すべてのメトリックは *Digger*（OpenTSDB）で使用でき、アプライアンス ID、コネクタ ID、およびルート範囲名で注釈が付けられます。さらに、コネクタサービスの Grafana ダッシュボードも、サービスからの重要なメトリックに使用できます。

コネクタおよび仮想アプライアンスの構成管理

構成の更新は、Cisco Secure Workload からアプライアンスとコネクタにプッシュできます。構成の更新を開始する前に、アプライアンスが Secure Workload に正常に登録され、アクティブになっている必要があります。同様に、コネクタは、コネクタサービスで構成の更新を開始する前に Secure Workload に登録されている必要があります。

アプライアンスとコネクタで可能な構成の更新には 3 つのモードがあります。

1. **テストと適用**：構成をテストし、テストが成功したら構成をコミットします。
2. **検出**：構成をテストし、テストが成功したら、構成に対して有効にできる追加のプロパティを検出します。
3. **削除**：構成を削除します。



（注） ERSPAN アプライアンスとコネクタは、構成の更新をサポートしていません。

テストおよび適用

テストおよび適用モードをサポートする設定では、目的のアプライアンスおよび/またはコネクタに設定を適用（確定）する前に設定を検証します。

NTP の設定（NTP Configuration）

NTP の設定により、アプライアンスは指定された NTP サーバーとクロックを同期できます。

パラメータ名	タイプ	説明
NTP を有効にする（Enable NTP）	チェックボックス	NTP 同期を有効にする必要がありますか?
[NTP Servers]	リスト ストリング	NTP サーバーのリスト。少なくとも 1 つのサーバーを指定する必要があります。最大 5 つのサーバーを指定できます。

テスト：ポート 123 で指定された NTP サーバーに対して UDP 接続を確立できるかどうかをテストします。いずれかの NTP サーバーでエラーが発生した場合は、この設定を受け入れないでください。

適用：/etc/ntp.conf を更新し、systemctl restart ntpd.service を使用して ntpd サービスを再起動します。*ntp.conf* を生成するためのテンプレートは次のとおりです。

```
# --- GENERAL CONFIGURATION ---
server <ntp-server>
...
server 127.127.1.0
fudge 127.127.1.0 stratum 10
# Drift file
driftfile /etc/ntp/drift
```

許可されている Cisco Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：なし

図 *118 : NTP* 設定のテスト中にエラーが発生

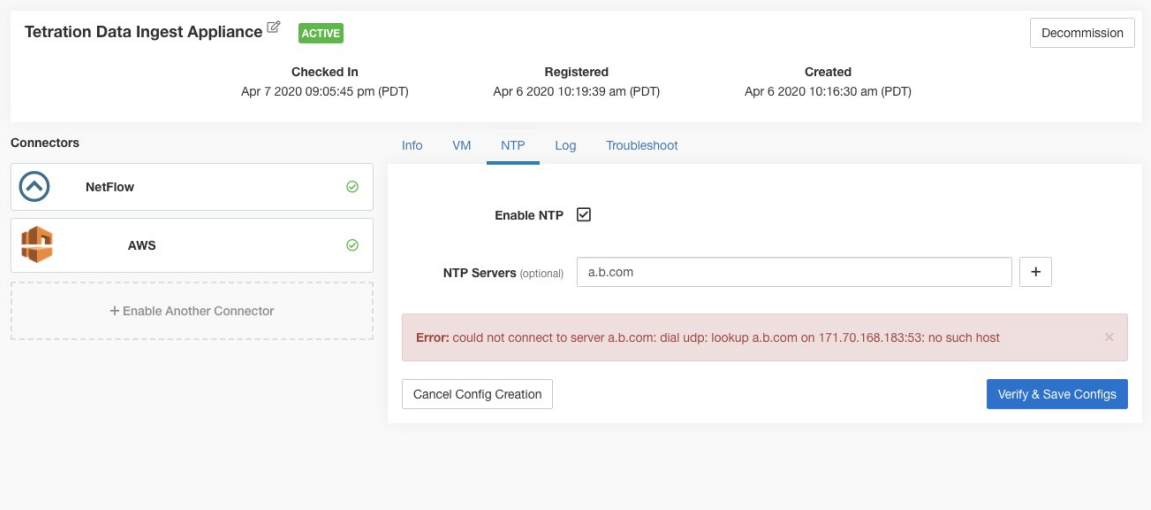


図 *119* : 有効な *NTP* サーバーを使用した *NTP* 設定

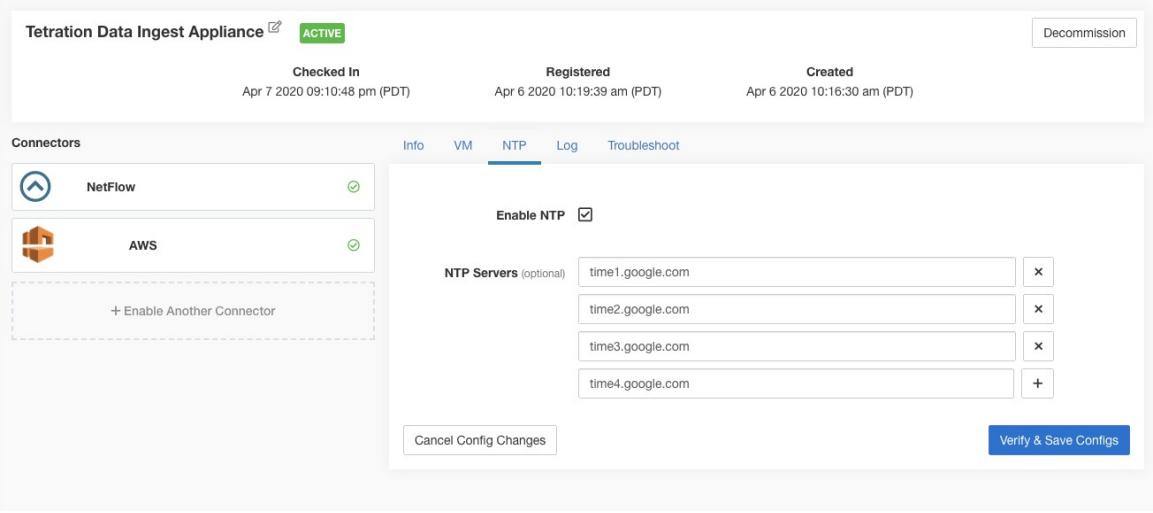
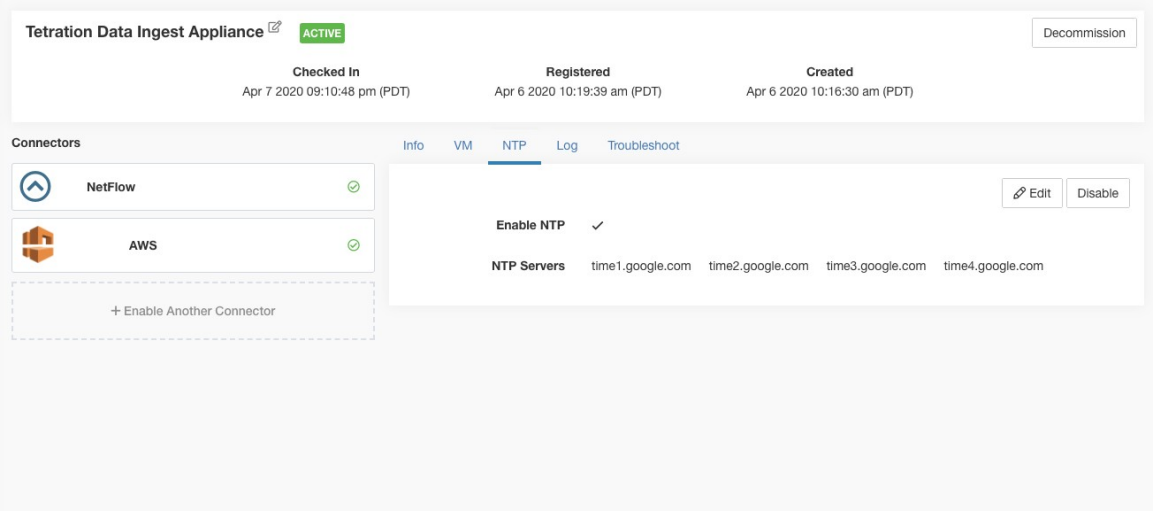


図 *120* : 検証および適用済みの *NTP* 設定



ログ設定

ログ構成により、アプライアンスやコネクタのログレベル、ログファイルの最大サイズ、ログローテーション パラメータが更新されます。アプライアンスで構成の更新がトリガーされると、アプライアンスコントローラのログ設定が更新されます。一方、コネクタで構成の更新がトリガーされると、サービスコントローラとサービスログの設定が更新されます。

パラメータ名	タイプ	説明
Logging level	dropdown	設定するロギングレベル
	• *debug*	デバッグログレベル
	• *info*	情報ログレベル
	• *warn*	警告ログレベル
	• *error*	エラーログレベル
[最大ログファイルサイズ（MB単位）（Max log file size (in MB)）]	number	ログ ローテーションが開始される前のログ ファイルの最大サイズ
[ログローテーション（日単位）（Log rotation (in days)）]	number	ログローテーションが開始されるまでのログ ファイルの最大経過時間
[ログローテーション（インスタンス単位）（Log rotation (in instances)）]	number	保持されるログファイルの最大インスタンス

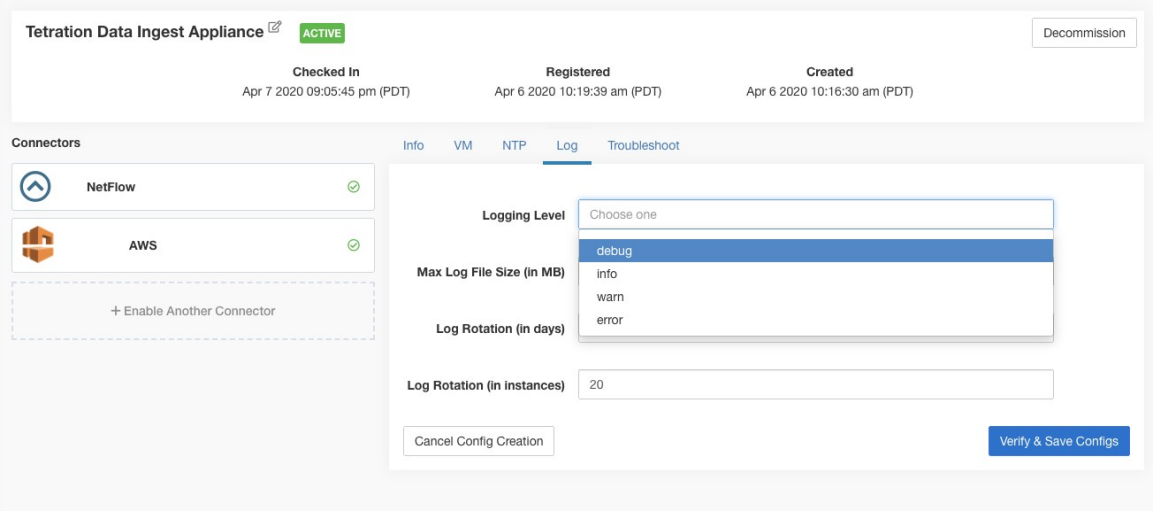
テスト：運用なし。

適用：アプライアンスで構成がトリガーされた場合は、アプライアンスの tet-controller の構成ファイルが更新されます。構成がコネクタでトリガーされた場合は、tet-controller の構成ファイルと、コネクタを管理する Docker コンテナ上のコントローラによって管理されるサービスが更新されます。

許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、ISE、ASA、Meraki。

図 *121* : アプライアンスのログ構成



（注） すべてのアラート通知コネクタ（Syslog、Email、Slack、PagerDuty、Kinesis）は Secure Workload Edge 上の単一の Docker サービス（Cisco Secure Workload Alert Notifier）で実行されるため、別のアラート通知コネクタの構成に影響を与えずにコネクタのログ構成を更新することはできません。Secure Workload Edge アプライアンス上の Secure Workload Alert Notifier（TAN）Docker サービスのログ構成は、許可されたコマンドを使用して更新できます。

詳細については、「アラート通知コネクタログ構成の更新」を参照してください。

エンドポイントの設定

エンドポイントの設定では、AnyConnect および ISE コネクタのエンドポイントの非アクティブタイムアウトを指定します。エンドポイントがタイムアウトすると、コネクタは Secure Workload とのチェックインを停止し、コネクタ上のエンドポイントのローカル状態を消去します。

パラメータ名	タイプ	説明
エンドポイントの非アクティブアイムアウト（分単位）（InactivityTimeout for Endpoints(in minutes)）	number	AnyConnect または ISE コネクタによって公開されたエンドポイントの非アクティブタイムアウト。タイムアウトすると、エンドポイントは Cisco Secure Workload をチェックインしなくなります。デフォルトは 30 分です。

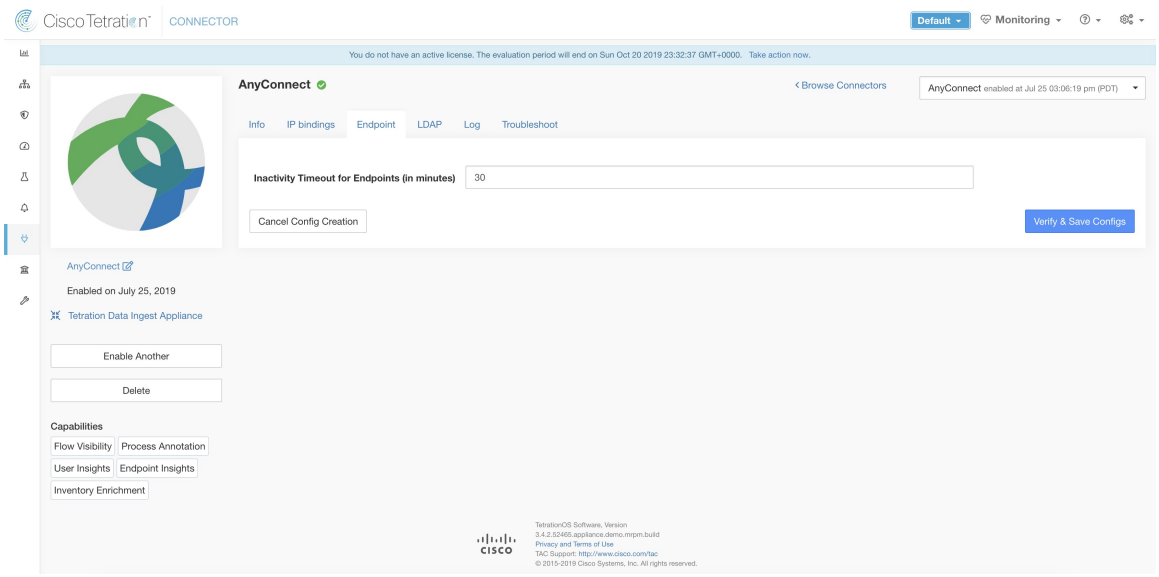
テスト：運用なし。

適用：新しい値でコネクタの構成ファイルを更新します。

許可されている Secure Workload 仮想アプライアンス：なし

許可されているコネクタ：AnyConnect および ISE

図 *122 : AnyConnect* コネクタのエンドポイント非アクティブタイムアウト設定



Slack 通知設定

Slack で Secure Workload アラートを公開するためのデフォルトの構成。

パラメータ名	タイプ	説明
Slack ウェブフック URL	string	Secure Workload アラートを発行する Slack ウェブフック

テスト：ウェブフックを使用してテストアラートを Slack に送信します。アラートが正常に投稿された場合、テストは合格です。

[適用（Apply）]：指定されたパラメータでコネクタの構成ファイルを更新します。

許可される Secure Workload 仮想アプライアンス：なし

許可されているコネクタ：Slack

PagerDuty 通知設定

PagerDuty で Secure Workload アラートを公開するためのデフォルト設定。

パラメータ名	タイプ	説明
PagerDuty サービスキー（PagerDuty Service Key）	string	PagerDuty で Cisco Secure Workload アラートをプッシュするための PagerDuty サービスキー

テスト：サービスキーを使用してテストアラートを PagerDuty に送信します。アラートが正常に発行された場合、テストは合格です。

[適用（Apply）]：指定されたパラメータでコネクタの構成ファイルを更新します。

許可される Secure Workload 仮想アプライアンス：なし

許可されたコネクタ：PagerDuty

Kinesis 通知設定

Amazon Kinesis で Secure Workload アラートを公開するためのデフォルトの設定。

パラメータ名	タイプ	説明
AWS アクセスキー ID（AWS Access Key ID）	string	AWS と通信するための AWS アクセスキー ID
AWS 秘密アクセスキー（AWS Secret Access Key）	string	AWS と通信するための AWS シークレットアクセスキー
AWS リージョン（AWS Region）	AWS リージョンのドロップダウン	Kinesis ストリームが設定されている AWS リージョンの名前
Kinesis ストリーム（Kinesis Stream）	string	Kinesis ストリームの名前
ストリームパーティション（Stream Partition）	string	ストリームのパーティション名

テスト：指定された設定を使用して、テストアラートを Kinesis ストリームに送信します。アラートが正常に発行された場合、テストは合格です。

[適用（Apply）]：指定されたパラメータでコネクタの構成ファイルを更新します。

許可される Secure Workload 仮想アプライアンス：なし

許可されているコネクタ：Kinesis

電子メール通知設定

電子メールで Secure Workload アラートを公開するためのデフォルト設定です。

パラメータ名	タイプ	説明
SMTP ユーザ名（SMTP Username）	string	SMTPサーバーのユーザー名。このパラメータはオプションです。
SMTP パスワード（SMTP Password）	string	ユーザーの SMTP サーバーパスワード（指定されている場合）。このパラメータはオプションです。
SMTP Server	string	SMTP サーバーの IP アドレスまたはホスト名
SMTP Port	number	SMTP サーバーのリスニングポート。デフォルト値は 587 です。
[Secure Connection]	チェックボックス	SMTP サーバー接続に SSL を使用する必要があるかどうか。
From Email Address	string	アラートの送信に使用する電子メールアドレス
[デフォルト受信者（Default Recipients）]	string	カンマで区切られた受信者の電子メールアドレス一覧。

[テスト（Test）]：指定された設定を使用してテスト電子メールを送信します。アラートが正常に発行された場合、テストは合格です。

[適用（Apply）]：指定されたパラメータでコネクタの構成ファイルを更新します。

許可される Secure Workload 仮想アプライアンス：なし

許可されるコネクタ：電子メール

Syslog 通知設定

Syslog で Secure Workload アラートを公開するためのデフォルト構成。

パラメータ名	タイプ	説明
Protocol	dropdown	サーバーへの接続に使用するプロトコル。
	• [UDP]	
	• [TCP]	
Server Address	string	Syslog サーバーの IP アドレスまたはホスト名。

パラメータ名	タイプ	説明
ポート（Port）	number	Syslog サーバーのリスニングポート。デフォルトのポート値は 514 です。

テスト：指定された構成を使用して、テストアラートをSyslogサーバーに送信します。アラートが正常に発行された場合、テストは合格です。

[適用（Apply）]：指定されたパラメータでコネクタの構成ファイルを更新します。

許可される Secure Workload 仮想アプライアンス：なし

許可されるコネクタ：Syslog

Syslog のシビラティ（重大度）のマッピング設定

次の表は、Syslog の Secure Workload アラートにおけるデフォルトのシビラティ（重大度）マッピングを示しています。

安全なワークロードアラートのシビラティ（重大度）	Syslog のシビラティ（重大度）
LOW	LOG_DEBUG
[中（Medium）]	LOG_WARNING
HIGH	LOG_ERR
CRITICAL	LOG_CRIT
即時対応（IMMEDIATE ACTION）	LOG_EMERG

この設定は、この構成を使用して変更できます。

パラメータ名	マッピングのドロップダウン
[即時対応（IMMEDIATE_ACTION）]	• *[緊急（Emergency）]* • *[アラート（Alert）]* • *[Critical]* • *[エラー（Error）]* • [警告（Warning）] • *[通知（Notice）]* • *[Informational]* • デバッグ（*Debug*）
CRITICAL	
HIGH	
[中（Medium）]	
LOW	

[テスト（Test）]：運用なし。

[適用（Apply）]：指定されたパラメータでコネクタの構成ファイルを更新します。

許可される Secure Workload 仮想アプライアンス：なし

許可されるコネクタ：Syslog

ISE インスタンスの設定

この構成では、Cisco Identity Services Engine（ISE）に接続するために必要なパラメータを指定します。この構成の複数のインスタンスを提供することにより、ISE コネクタは複数の ISE アプライアンスに接続してエンドポイントに関するメタデータをプルできます。最大 20 のISE 構成インスタンスを提供できます。

パラメータ名	タイプ	説明
ISEクライアント証明書（ISE Client Certificate）	string	pxGrid を使用して ISE に接続するための ISE クライアント証明書
ISEクライアントキー（ISE Client Key）	string	ISE に接続するための ISE クライアントキー
ISEサーバーCA証明書（ISE Server CA Certificate）	string	ISE の CA 証明書
ISE のホスト名	string	ISE pxGrid の FQDN
ISEノード名（ISE Nodename）	string	ISE pxGrid のノード名

テスト：指定されたパラメータを使用して ISE に接続します。接続に成功したら、構成を受け入れます。

[適用（Apply）]：指定されたパラメータでコネクタの構成ファイルを更新します。

許可される Secure Workload 仮想アプライアンス：なし

許可されているコネクタ：ISE

検出

検出モードをサポートする設定では、次のことが行われます。

1. ユーザーから基本設定を収集します。
2. 基本設定を検証します。
3. 設定に関する追加のプロパティを検出してユーザーに提示します。
4. 検出されたプロパティを使用して、ユーザーが設定を拡張できるようにします。

5. 拡張された設定を検証して適用します。

3.3.1.x リリースでは、LDAP 構成で検出モードがサポートされています。

LDAP 設定

LDAP設定では、LDAPへの接続方法、使用する基本識別名（DN）、ユーザー名に対応する属性、およびユーザー名ごとにフェッチする属性を指定します。LDAP属性は、その環境に固有のLDAPのプロパティです。

LDAPへの接続方法と基本DNの設定があれば、LDAP内のユーザーの属性を検出できます。これらの検出された属性は、UIでユーザーに表示できます。ユーザーは、検出されたこれらの属性から、ユーザー名に対応する属性と、ユーザー名ごとにLDAPから収集する最大6つの属性のリストを選択します。その結果、LDAP属性を手動で設定する必要がなくなり、エラーが減少します。

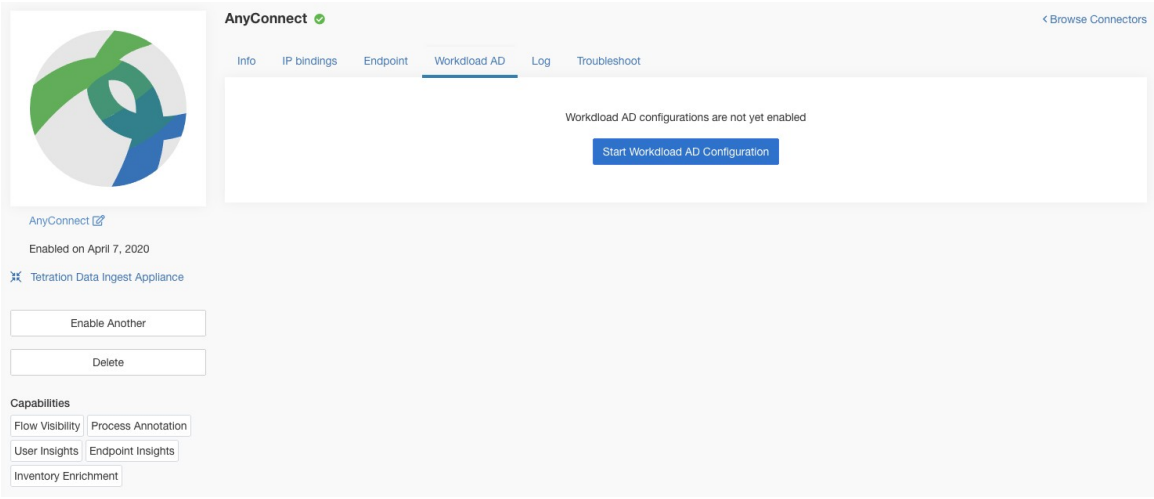
検出を通じてLDAP設定を作成するための詳細な手順は次のとおりです。

手順

ステップ1 LDAP設定の開始

コネクタのLDAP設定を開始します。

図 *123 : LDAP* 設定の検出の開始



ステップ2 基本のLDAP設定の指定

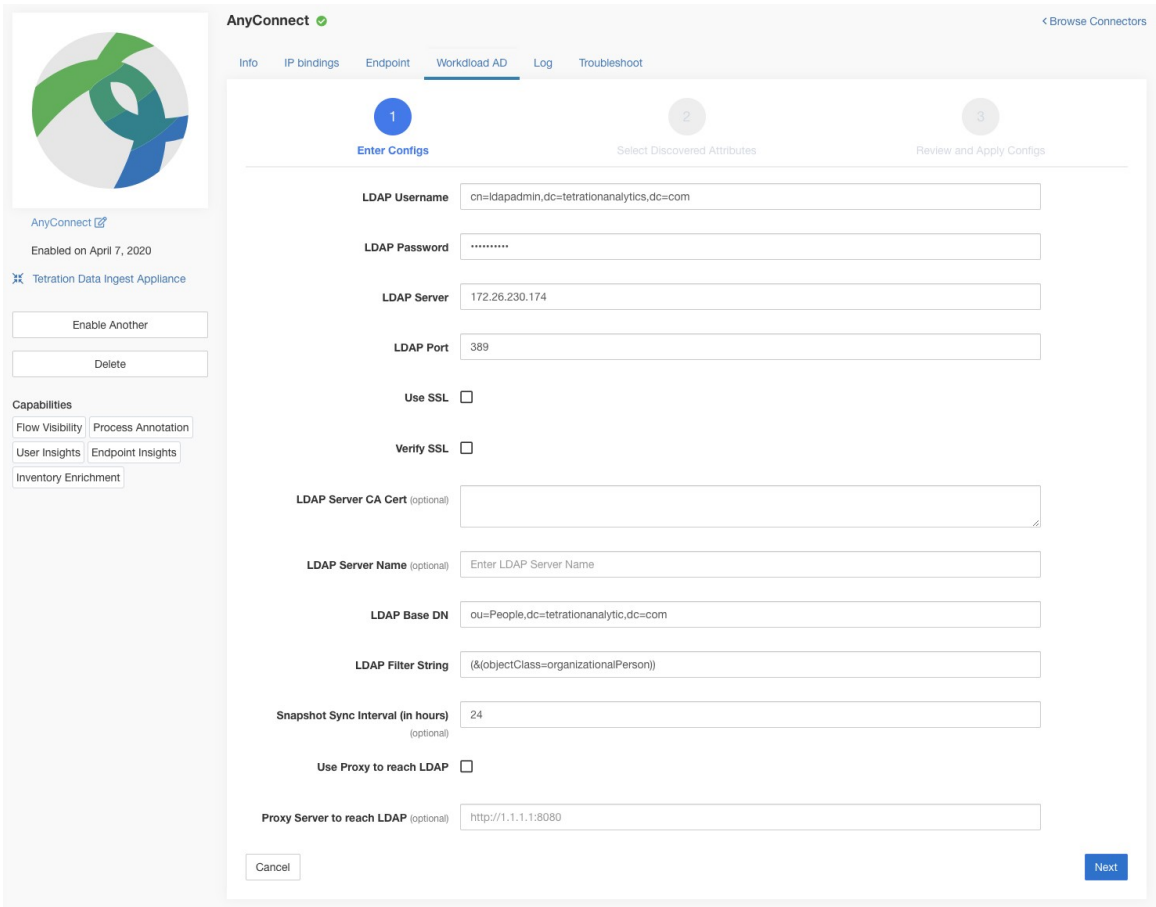
LDAPに接続するための基本設定を指定します。この設定では、ユーザーはLDAPサーバーに接続するためのLDAPバインドDNまたはユーザー名、LDAPサーバーへの接続に使用するLDAPパスワード、LDAPサーバーのアドレス、LDAPサーバーのポート、接続するベースDN、およびこのフィルタに一致するユーザーをフェッチするためのフィルタ文字列を提供します。

パラメータ名	タイプ	説明
LDAP ユーザ名（LDAP Username）	string	LDAP サーバーにアクセスするための LDAP ユーザー名またはバインド DN*
LDAPパスワード（LDAP Password）	string	LDAP サーバーにアクセスするためのユーザー名の LDAP パスワード*
LDAP サーバー（LDAP Server）	string	LDAP サーバーアドレス
[LDAPポート（LDAP Port）]	number	LDAP サーバーポート
SSL を使用する（Use SSL）	チェックボックス	コネクタは LDAP に安全に接続する必要がありますか。オプション。デフォルトは false です。
[SSLの確認（Verify SSL）]	チェックボックス	コネクタは LDAP 証明書を検証する必要がありますか。オプション。デフォルトは false です。
[LDAPサーバーCA証明書（LDAP Server CA Cert）]	string	サーバーCA証明書。オプション。
LDAP Server Name	string	LDAP証明書が発行されるサーバー名（SSL の検証がチェックされている場合は必須です）。
LDAP Base DN	string	LDAP 基本 DN、LDAP でのディレクトリ検索の開始点
[LDAPフィルタ文字列（LDAP Filter String）]	string	LDAPフィルタのプレフィックス文字列。この条件のみに一致する検索結果をフィルタ処理します。
[スナップショットの同期間隔（時間単位）（Snapshot Sync Interval (in hours)）]	number	LDAP スナップショットを（再）作成する時間間隔を時間単位で指定します。オプション。デフォルトは 24 時間です。

パラメータ名	タイプ	説明
[プロキシを使用してLDAPにアクセス（Use Proxy to reach LDAP）]	チェックボックス	コネクタは LDAP サーバーにアクセスするためにプロキシサーバーを使用する必要がありますか。
[LDAPにアクセスするためのプロキシサーバー（Proxy Server to reach LDAP）]	string	LDAP にアクセスするためのプロキシサーバー

コネクタで LDAP を設定するために必要な最小限のユーザー権限は、**標準ドメインユーザー**です。

図 *124* : 初期 *LDAP* 設定

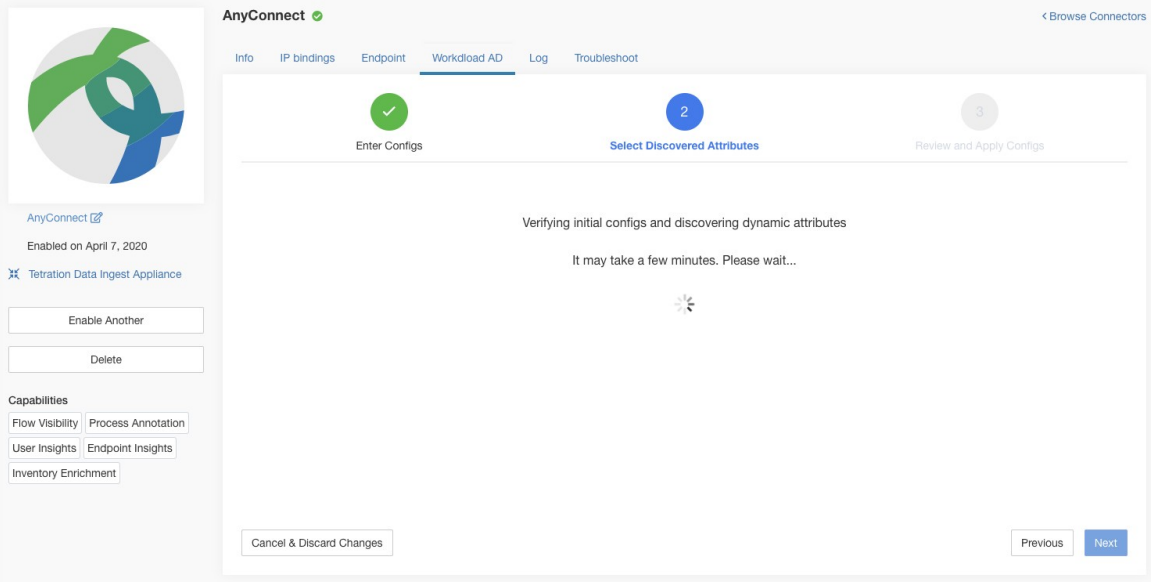


ステップ 3　検出処理中

ユーザーが [次へ（Next）] をクリックすると、この設定がコネクタに送信されます。コネクタは、指定された設定を使用して LDAP サーバーとの接続を確立します。LDAP サーバーから最大 1000 人のユーザーをフェッチし、すべての属性を識別します。さらに、1000 人のユーザー

すべてに共通するすべての単一値属性のリストを計算します。コネクタは、この結果を Cisco Secure Workload に返します。

図 *125 : Discovery in progress*



ステップ 4　検出された属性で設定を強化

ユーザーは、ユーザー名に対応する属性を選択し、組織内の各ユーザー（フィルタ文字列に一致するユーザー）について、コネクタがフェッチおよびスナップショットする必要がある最大 6 つの属性を選択する必要があります。このアクションは、検出された属性のリストのドロップダウンを使用して実行されます。そのため、手動エラーや設定ミスが削減されます。

パラメータ名	タイプ	説明
[LDAPユーザー名属性（LDAP Username Attribute）]	string	ユーザー名を含む LDAP 属性
[フェッチするLDAP属性（LDAP Attributes to Fetch）]	文字列のリスト	ユーザーに対して取得する必要がある LDAP 属性のリスト

図 *126* : 検出された *LDAP* 属性

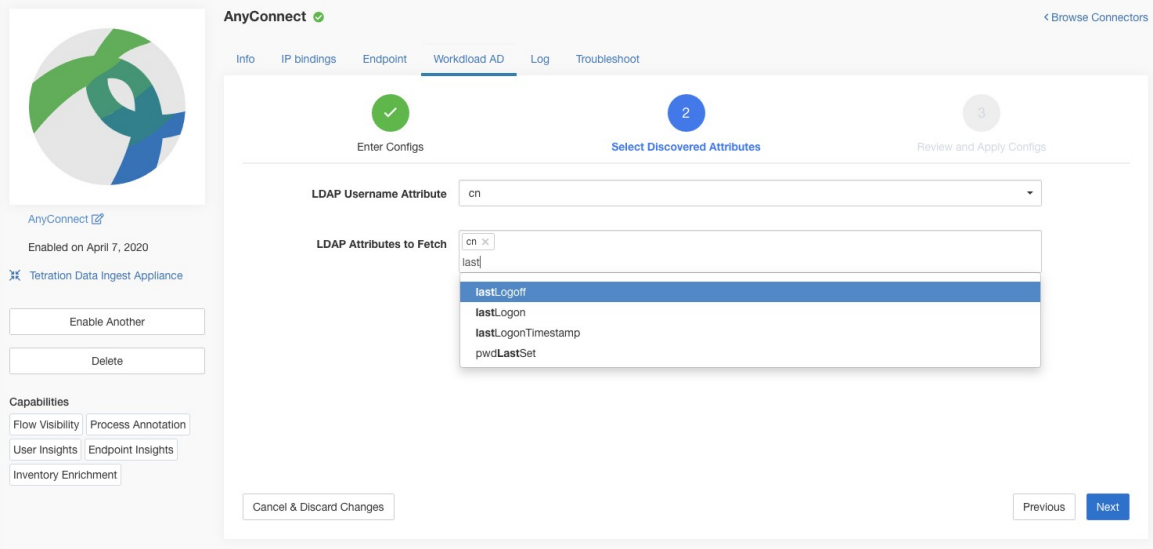
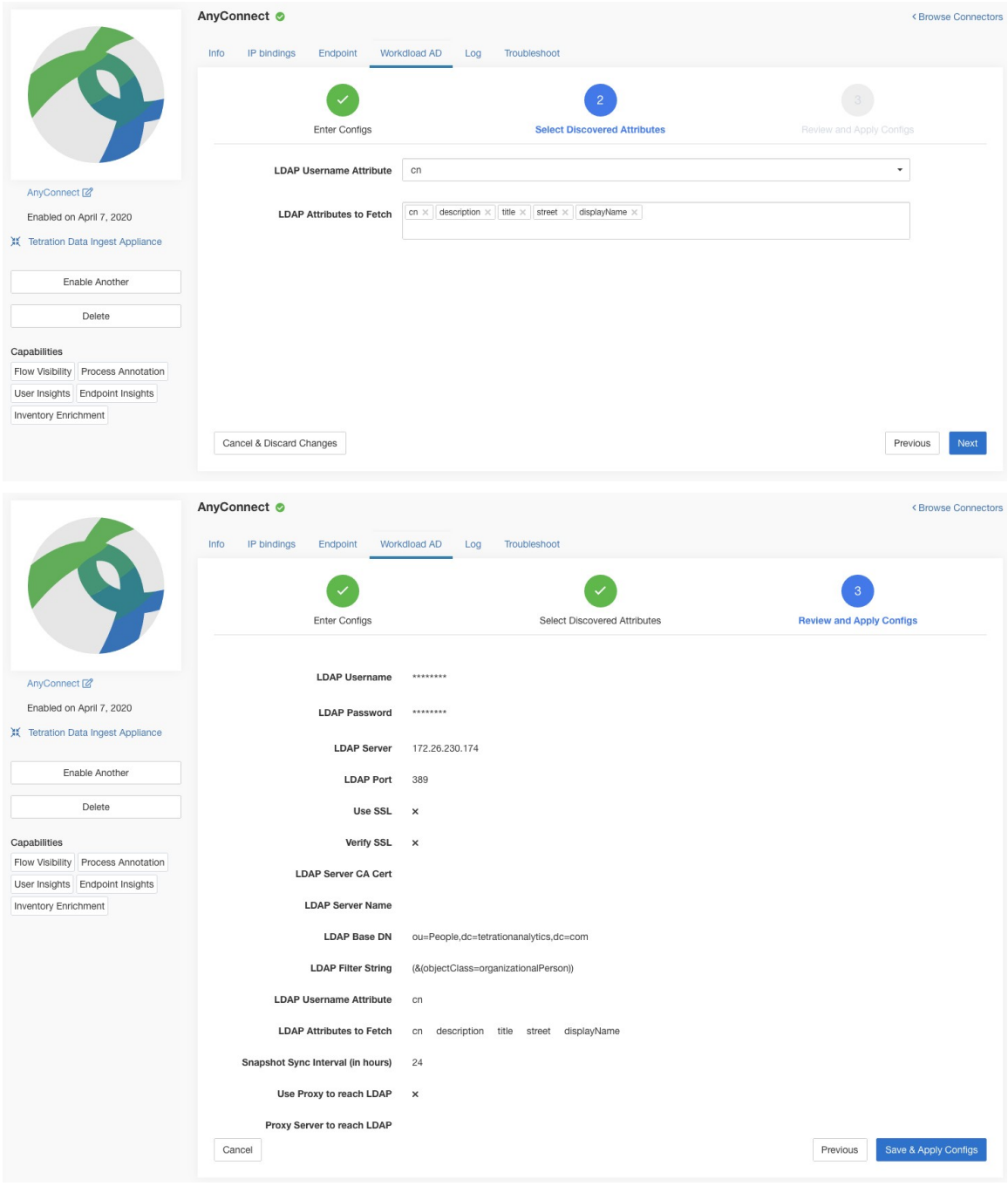


図 *127* : ユーザー名属性およびユーザー名ごとに収集する属性を特定

ステップ 5 設定の終了、保存、および適用

最後に [変更を保存して適用（Save and Apply Changes）] をクリックすると、設定が完了します。

図 *128 : LDAP* 設定の検出の完了とコミット



コネクタは、完了した設定を受信します。フィルタ文字列に一致するすべてのユーザーのローカルスナップショットを作成し、選択した属性のみをフェッチします。スナップショットが完了すると、インベントリ内のユーザーとそのLDAP属性に注釈を付けるため、コネクタサービスによるスナップショットの使用を開始できます。

許可される Secure Workload 仮想アプライアンス：なし

許可されるコネクタ：AnyConnect、ISE、F5

削除（Remove）

追加されたすべての設定は、コネクタやアプライアンスから削除できます。それぞれの設定には、ユーザーが設定を削除できる [削除（Delete）] ボタンがあります。

トラブルシューティング

コネクタと仮想アプライアンスは、考えられる問題をデバッグするためのさまざまなトラブルシューティング メカニズムをサポートしています。

（注） このセクションは次のものには該当しません。

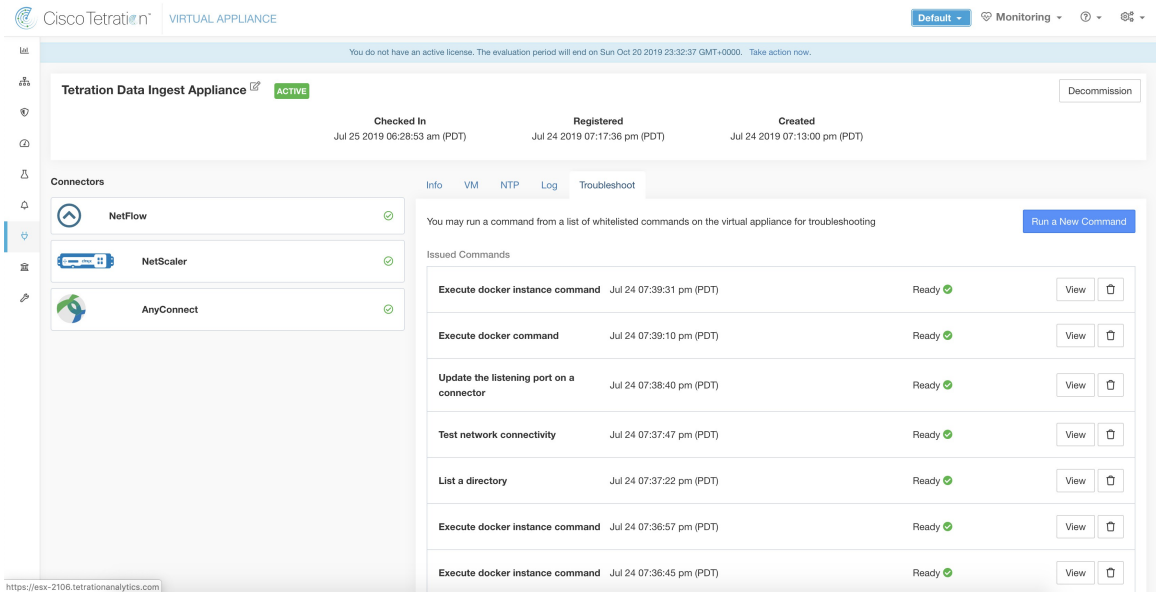
ERSPAN仮想アプライアンス：トラブルシューティングの詳細については、ERSPANアプライアンスのページを参照してください。

クラウドコネクタ：クラウドコネクタのトラブルシューティングを行うには、クラウドコネクタのセクションを参照してください（「AWS コネクタに関する問題のトラブルシューティング」など）。

許可されている一連のコマンド

許可されている一連のコマンドにより、アプライアンスおよびDockerコンテナ（コネクタ用）で一部のデバッグコマンドを実行できるようになります。許可されているコマンドには、ログと現在の実行コンフィギュレーションを取得する機能、ネットワーク接続をテストする機能、指定されたポートに一致するパケットをキャプチャする機能が含まれます。

図 *129 : Secure Workload* 仮想アプライアンスのトラブルシュートページ



（注） 許可されているコマンドセットを使用したトラブルシューティングは、カスタマーサポートロールを持つユーザーのみがアプライアンスとコネクタで利用できます。

Show Logs

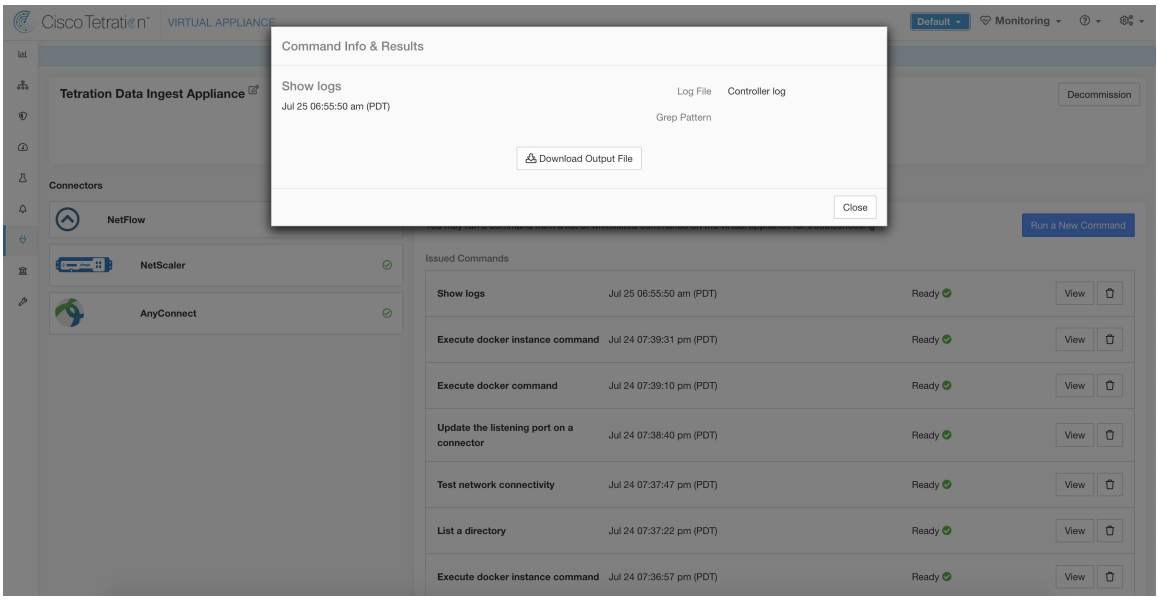
コントローラログファイルの内容を表示し、オプションで指定されたパターンのファイルを grep します。Secure Workload は、コマンドが発行されたアプライアンスまたはコネクタにコマンドを送信します。アプライアンスまたはコネクタサービスのコントローラから結果が返されます（最後の 5000 行の末尾）。結果が Cisco Secure Workload で利用可能になると、ファイルをダウンロードするためのダウンロードボタンが表示されます。

引数名	タイプ	説明
[Grepパターン（Grep Pattern）]	string	ログファイルから grep するパターン文字列

許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、E メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *130 : Secure Workload* 入力アプライアンスからの *Show Logs* 出力のダウンロード



サービスログを表示

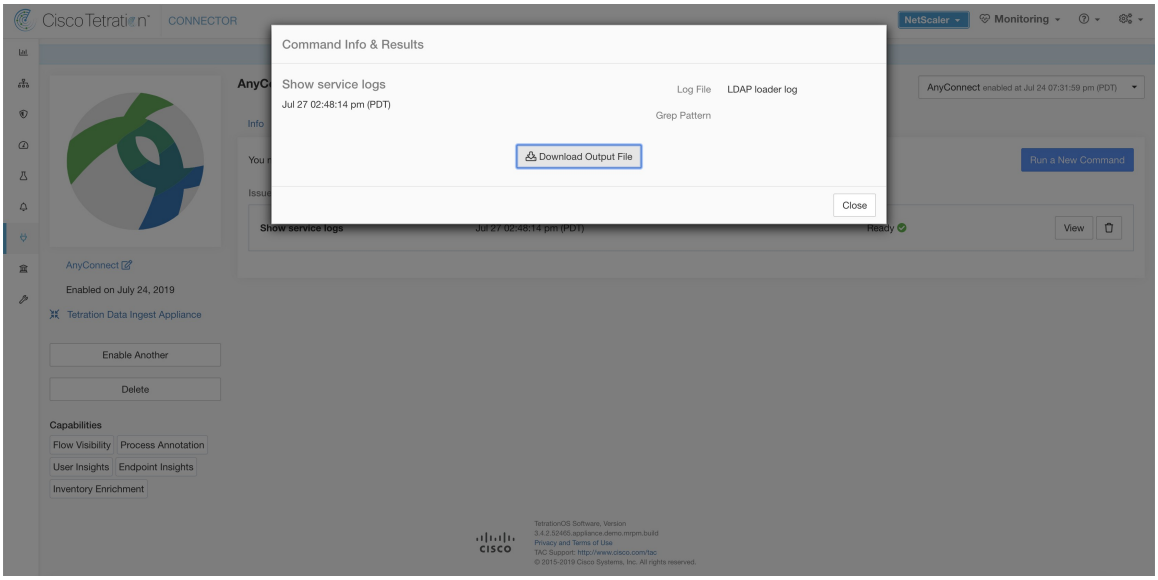
サービスログファイルの内容を表示し、オプションでファイルをgrep して指定されたパターンを見つけます。Secure Workloadは、コマンドが発行されたアプライアンスまたはコネクタにコマンドを送信します。アプライアンスまたはコネクタサービスのコントローラから結果が返されます（最後の 5000 行の末尾）。結果が Cisco Secure Workload で利用可能になると、ファイルをダウンロードするためのダウンロードボタンが表示されます。

引数名	タイプ	説明
Log File	dropdown	収集するログファイルの名前
	• サービスログ（Service log）	コネクタサービスのログ
	• アップグレードログ（Upgrade log）	サービスのアップグレードログ
	• LDAPローダーログ（LDAP loader log）	LDAP が有効になっているコネクタの LDAP スナップショットのログ
[Grepパターン（Grep Pattern）]	string	ログファイルから grep するパターン文字列

許可された Secure Workload 仮想アプライアンス：なし（有効なコネクタサービスでのみ利用可能）

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、電子メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *131 : LDAP* ローダーログファイルを対象とした *AnyConnect* コネクタからの *[*サービスログを表示（*Show Service Logs*）*]* 出力のダウンロード



show running-config の出力結果

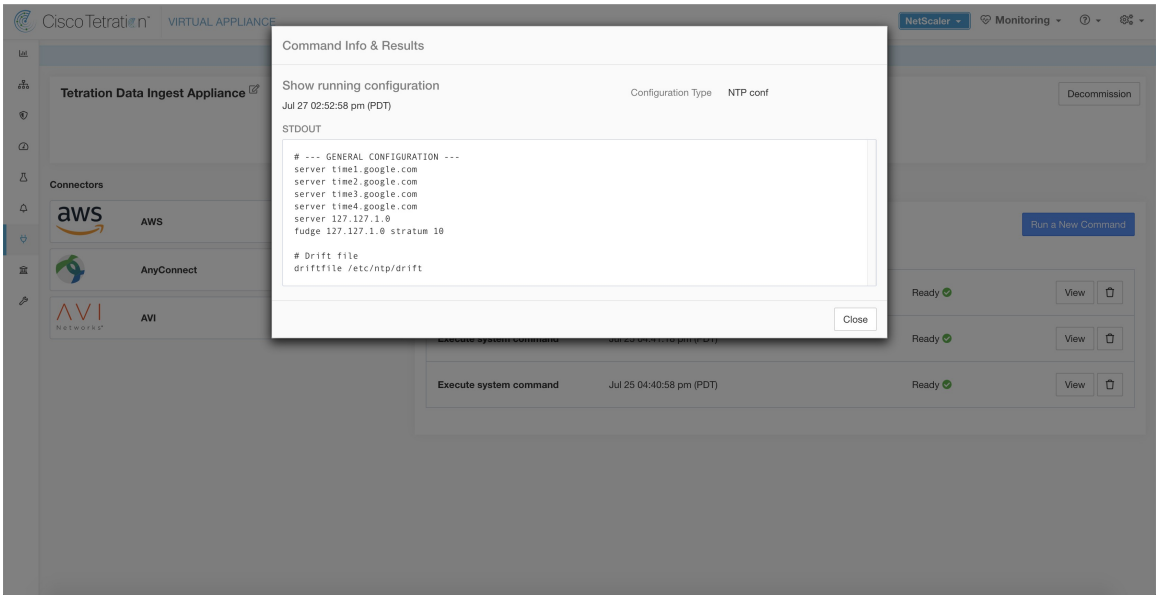
アプライアンスまたはコネクタのコントローラの実行コンフィギュレーションを表示します。アプライアンスまたはコネクタのコントローラが、要求された引数に対応する構成を取得し、結果を返します。Cisco Secure Workload で結果が得られると、設定の内容がテキストボックスに表示されます。

引数名	タイプ	説明
設定の種類	dropdown	収集する構成ファイル
	• コントローラ構成	アプライアンスコントローラの構成ファイル
	• スーパーバイザ構成	コントローラを実行するスーパーバイザの構成ファイル
	• NTP 構成	NTP 構成ファイル

許可されている Secure Workload 仮想アプライアンス: すべて

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、E メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *132 : Secure Workload Ingest* アプライアンスの *NTP* 構成の実行コンフィギュレーションの表示



サービス実行設定の表示

アプライアンスのコネクタ用にインスタンス化されたサービスの実行設定を表示します。サービスのコントローラは、要求された引数に対応する設定を取得し、結果を返します。Cisco Secure Workload で結果が得られると、設定の内容がテキストボックスに表示されます。

引数名	タイプ	説明
設定の種類	dropdown	収集する設定ファイル。
	• コントローラ設定	サービスコントローラの設定ファイル
	• スーパーバイザ設定	コントローラを実行するスーパーバイザの設定ファイル。
	• サービス設定	サービス設定ファイル
	• LDAP 設定	LDAP が有効化されているコネクタの LDAP 設定。

許可された Secure Workload 仮想アプライアンス：なし（有効なコネクタサービスでのみ利用可能）

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、電子メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

システムコマンドの表示

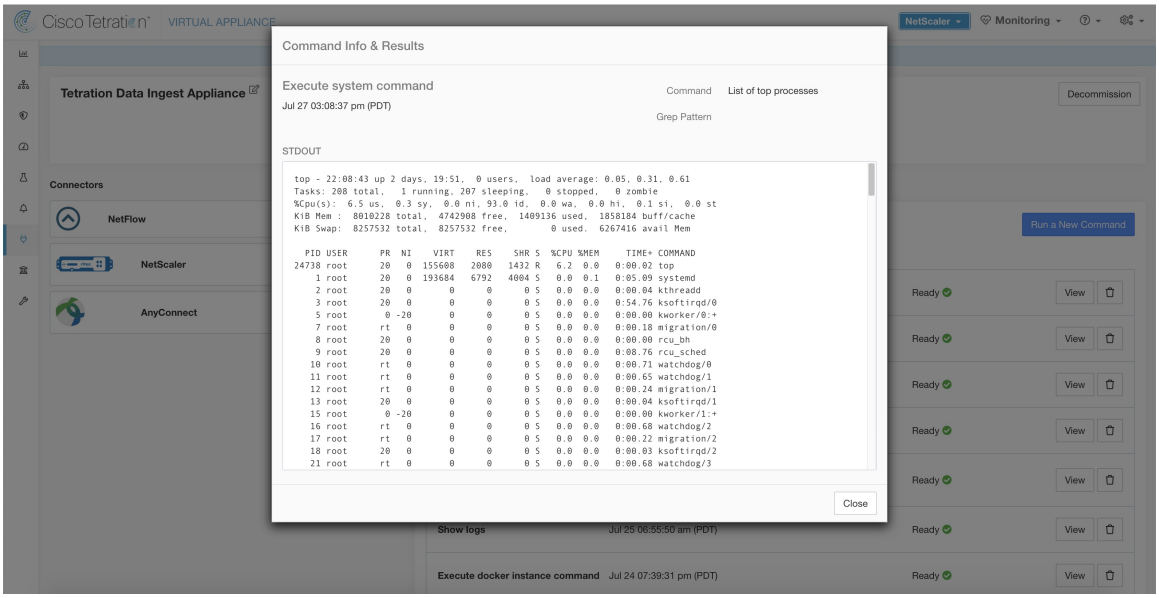
システムコマンドを実行し、指定したパターンの grep を実行します（オプション）。アプライアンスまたはコネクタサービスのコントローラから結果が返されます（最後の 5000 行の末尾）。必要に応じて、grep パターンを引数として指定でき、そのパターンに応じて出力がフィルタリングされます。結果が Cisco Secure Workload で利用可能になると、テキストボックスに表示されます。

引数名	タイプ	説明
システムコマンド	dropdown	実行するシステムコマンド
	• *IP* 設定（*IP configuration*）	**ifconfig**
	• IP ルート構成（IP route configuration）	**ip route**
	• IP パケットフィルタリングルール（IP packet filtering rules）	**iptables -L**
	• *Network* ステータス	**netstat**
	• プロセスステータス（Process status）	**ps -aux**
	• 上位プロセスのリスト（List of top processes）	**top -b -n 1**
	• NTP ステータス（NTP status）	**ntpstat**
	• NTP クエリ（NTP query）	**ntpq -pn**
	• CPU 情報（CPU info）	**lscpu**
	• メモリ情報（Memory info）	**lsmem**
	• ディスク空き領域（Disk free）	**df -H**
[Grepパターン（Grep Pattern）]	string	出力から grep するパターン文字列

許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、E メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *133 : Secure Workload Ingest* アプライアンスにシステムコマンドを表示して、上位プロセスのリストを取得する



Docker コマンドの表示

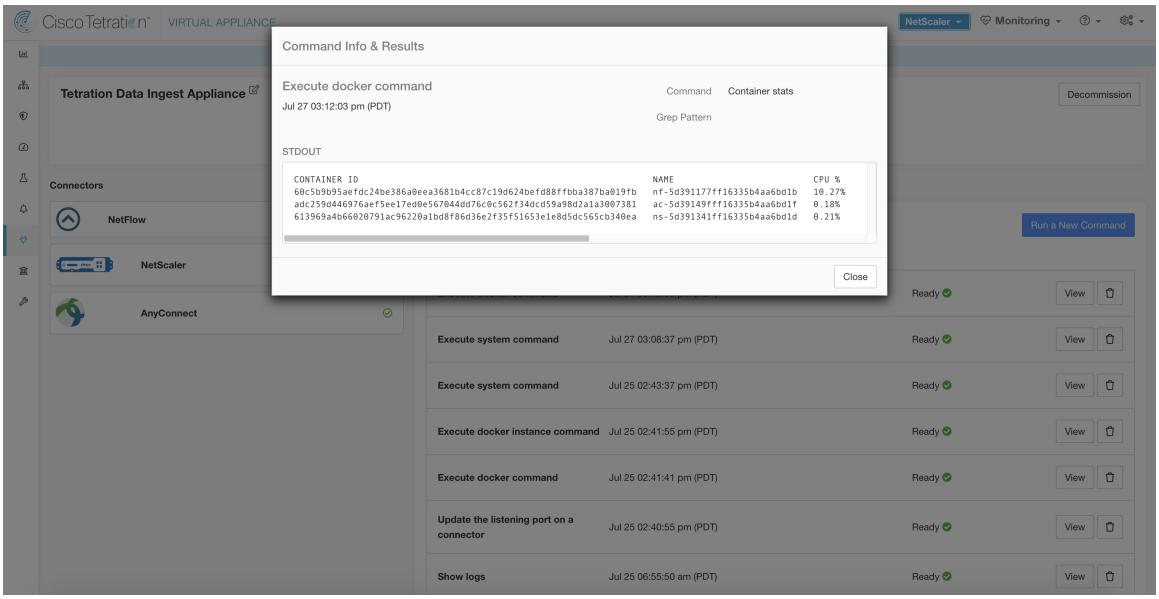
Docker コマンドを実行し、指定したパターンの grep を実行します（オプション）。コマンドは、アプライアンスコントローラによってアプライアンス上で実行されます。結果は、最後の 5000 行まで表示されます。必要に応じて、grep パターンを引数として指定でき、そのパターンに応じて出力がフィルタリングされます。結果が Cisco Secure Workload で利用可能になると、テキストボックスに表示されます。

引数名	タイプ	説明
Docker コマンド	dropdown	実行する Docker コマンド
	• Docker 情報（Docker info）	**docker info**
	• イメージの一覧表示（List images）	**docker images --no-trunc**
	• コンテナの一覧表示（List containers）	**docker ps --no-trunc**
	• ネットワークの一覧表示（List networks）	**docker network ls --no-trunc**
	• ボリュームの一覧表示（List networks）	**docker volume ls**
	• コンテナの統計情報（Container stats）	**docker stats --no-trunc--no-stream**
	• Docker ディスク使用率（Docker disk usage）	`docker system df -v`
	• Docker システムイベント（Docker system events）	**docker system events --since '10m'**
	• バージョン	**docker version**
[Grepパターン（Grep Pattern）]	string	出力から grep するパターン文字列

許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：なし

図 *134 : Secure Workload Ingest* アプライアンスで *Docker* コマンドを実行して、コンテナの統計情報を表示する



Docker インスタンスコマンドの表示

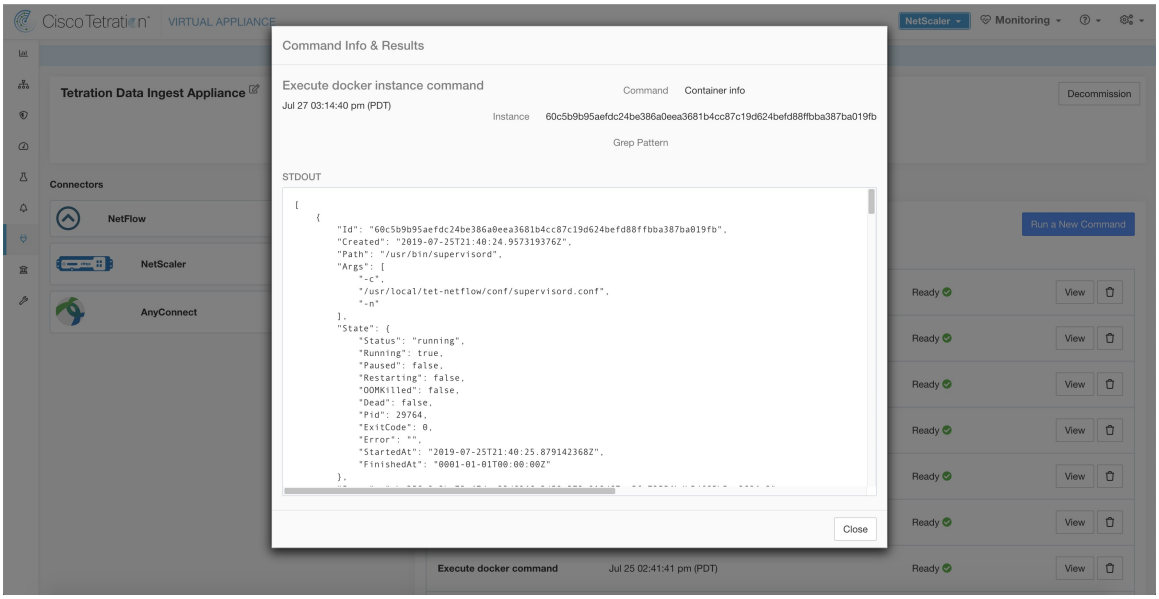
Docker リソースの特定のインスタンスで Docker コマンドを実行します。インスタンス ID は、Docker コマンドの表示を使用して取得できます。コマンドは、アプライアンスコントローラによってアプライアンス上で実行されます。結果は、最後の 5000 行まで表示されます。必要に応じて、grep パターンを引数として指定でき、そのパターンに応じて出力がフィルタリングされます。結果が Cisco Secure Workload で利用可能になると、テキストボックスに表示されます。

引数名	タイプ	説明
Docker コマンド	dropdown	実行する Docker コマンド
	• イメージ情報	**docker images --no-trunc <instance>**
	• ネットワーク情報	**docker network inspect <instance>**
	• ボリューム情報	**docker volume inspect <instance>**
	• コンテナ情報	**docker container inspect--size <instance>**
	• コンテナログ	**docker logs --tail 5000 <instance>**
	• コンテナポートのマッピング	**docker port <instance>**
	• コンテナリソース使用状況の統計	**docker stats --no-trunc--no-stream <instance>**
	• コンテナ実行プロセス	**docker port <instance>**
インスタンス	string	Docker リソース（イメージ、ネットワーク、ボリューム、コンテナ）ID（「Docker コマンドの表示」を参照）
[Grepパターン（Grep Pattern）]	string	出力から grep するパターン文字列

許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：なし

図 *135 : Secure Workload Ingest* アプライアンスで *Docker* インスタンスコマンドを実行して、コンテナ情報を取得する



Supervisor コマンドの表示

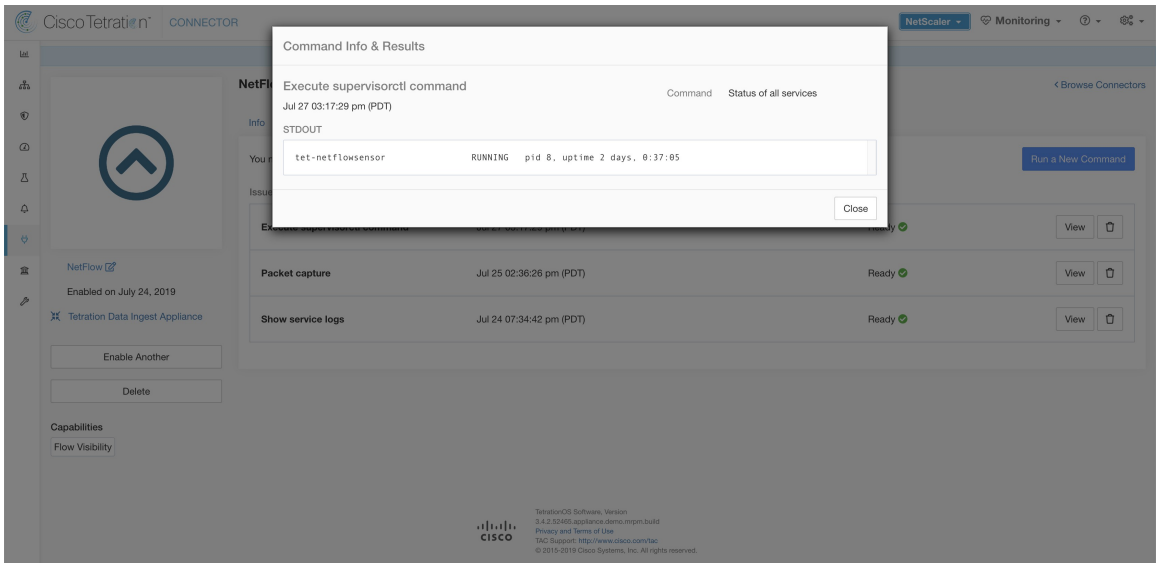
supervisorctl コマンドを実行し、結果を返します。Secure Workload はコマンドが発行されたアプライアンスまたはコネクタにコマンドを送信します。アプライアンスまたはコネクタサービスのコントローラから結果が返されます。結果は、Cisco Secure Workload で利用可能な場合はテキストボックスに表示されます。

引数名	タイプ	説明
SupervisorCtl コマンド	dropdown	実行する *supervisorctl* コマンド
	• 全サービスのステータス	**supervisorctl ステータス**
	• スーパーバイザの PID	**supervisorctl pid**
	• 全サービスの PID	**supervisorctl pid all**

許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、E メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *136* : 全サービスのステータスを取得するために *NetFlow* コネクタで *supervisorctl* コマンドを実行

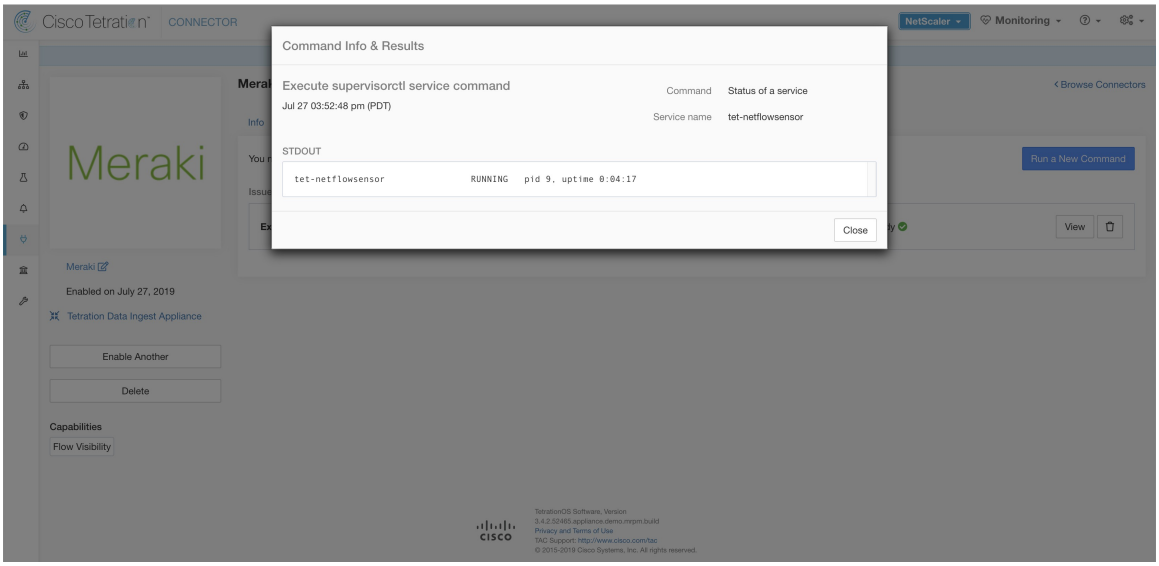


スーパーバイザサービスのコマンドの表示

特定のサービスで supervisorctl コマンドを実行します。サービス名は Supervisor コマンドの表示 を使用して取得できます。 Secure Workload は、コマンドが発行されたアプライアンスまたはコネクタにコマンドを送信します。アプライアンスまたはコネクタサービスのコントローラが結果を返します。結果は、Cisco Secure Workload で利用可能な場合はテキストボックスに表示されます。

引数名	タイプ	説明
SupervisorCtl コマンド	dropdown	実行する *supervisorctl* コマンド
	• サービスの状態	**supervisorctl status <service name>**
	• サービスのPID	**supervisorctl pid <service name>**
[サービス名（Service Name）]	string	スーパーバイザが制御するサービス名（「Supervisor コマンドの表示」を参照）。

図 *137 : NetFlow* コネクタで *supervisorctl* コマンドを実行し、指定されたサービス名のステータスを取得



許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、E メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

ネットワーク接続コマンド

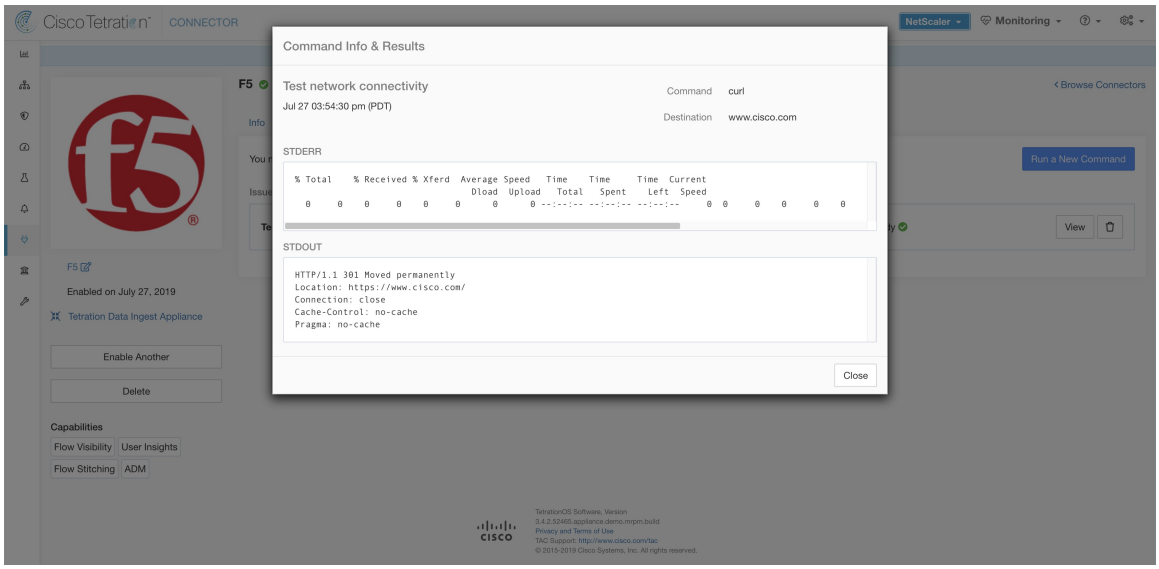
アプライアンスやコネクタからネットワーク接続をテストします。コマンドは、アプライアンスコントローラによってアプライアンス上で実行されます。結果が Cisco Secure Workload で利用可能になると、テキストボックスに表示されます。

引数名	タイプ	説明
ネットワークコマンド（Network Command）	dropdown	実行するネットワーク接続コマンド
	• *ping*	**ping -c 5 <destination>**
	• *curl*	**curl -I <destination>**
接続先（Destination）	string	テストに使用する接続先

許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、E メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *138 : curl* の実行による *F5* コネクタのネットワーク接続テスト



ファイルの一覧表示

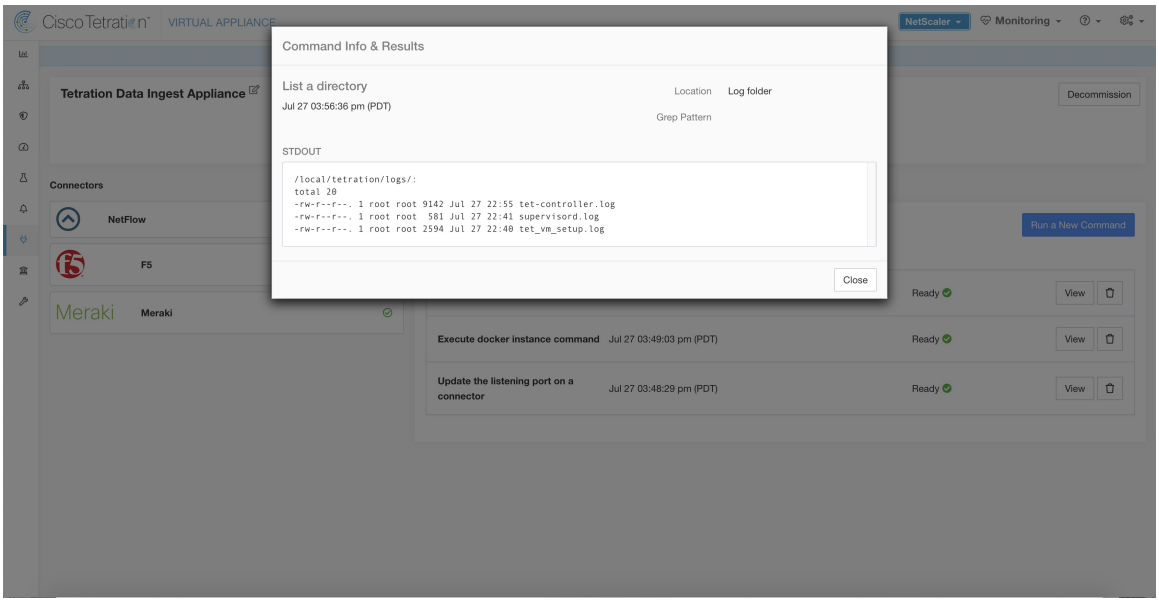
アプライアンスの既知の場所にあるファイルを一覧表示します。（オプション）指定されたパターンの grep。Secure Workload は、コマンドが発行されたアプライアンスにコマンドを送信します。アプライアンスのコントローラが結果を返します。結果が Cisco Secure Workload で利用可能になると、テキストボックスに表示されます。

引数名	タイプ	説明
所在地（**Location**）	dropdown	対象の場所にあるファイルの一覧表示
	• コントローラ構成フォルダ	コントローラ構成ファイルが保存されているフォルダの内容を一覧表示します。
	• コントローラ証明書フォルダ	コントローラ証明書が保存されているフォルダの内容を一覧表示します。
	• ログフォルダ（Log folder）	ログファイルが存在するフォルダの内容を一覧表示します。
[Grepパターン（Grep Pattern）]	string	出力から grep するパターン文字列

許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：なし

図 *139 : Secure Workload Ingest* アプライアンスのログフォルダ内のファイルを一覧表示



サービスファイルの一覧表示

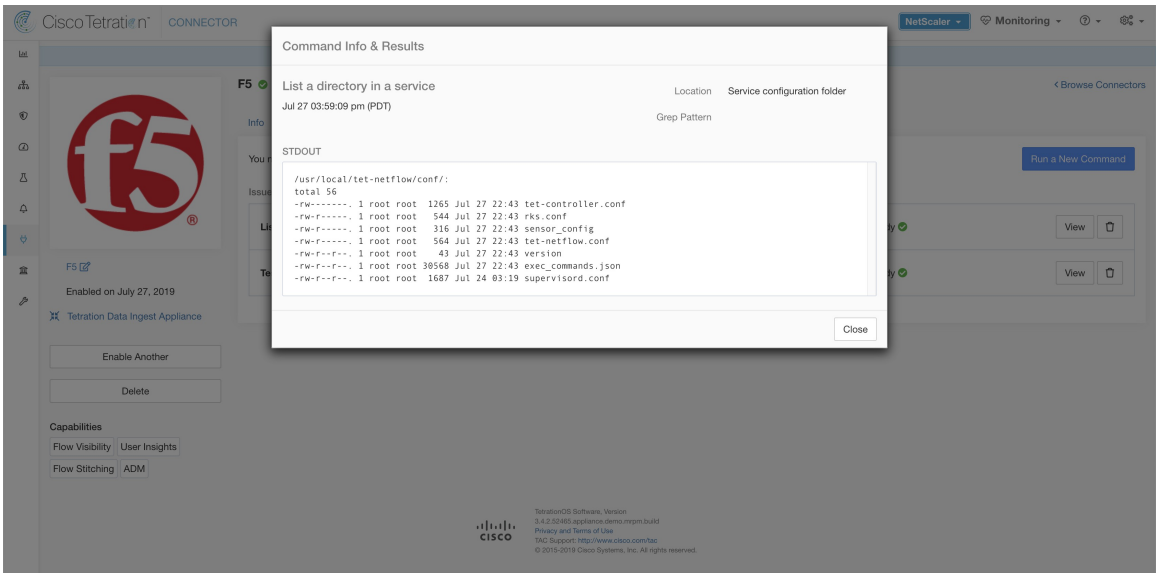
コネクタサービスの既知の場所にあるファイルを一覧表示します。（オプション）指定されたパターンの grep。Secure Workload は、コマンドが発行されたコネクタにコマンドを送信します。コネクタサービスのコントローラが結果を返します。結果が Cisco Secure Workload で利用可能になると、テキストボックスに表示されます。

引数名	タイプ	説明
所在地（**Location**）	dropdown	対象の場所にあるファイルを一覧表示します。
	• サービス コンフィギュレーション フォルダ（Service configuration folder）	サービス コンフィギュレーション ファイルが保存されているフォルダの内容を一覧表示します。
	• サービス証明書フォルダ（Service cert folder）	サービス証明書が保存されているフォルダの内容を一覧表示します。
	• ログフォルダ（Log folder）	ログファイルが存在するフォルダの内容を一覧表示します。
	• DBフォルダ（DB folder）	エンドポイント（特に AnyConnect および ISE コネクタ）の状態が保持されているフォルダの内容を一覧表示します。
[Grepパターン（Grep Pattern）]	string	出力から grep するパターン文字列

許可されている Secure Workload 仮想アプライアンス：なし

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、Email、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *140 : Secure Workload Ingest* アプライアンスの *F5* コネクタのコンフィギュレーションフォルダ内ファイルを一覧表示



パケットキャプチャ

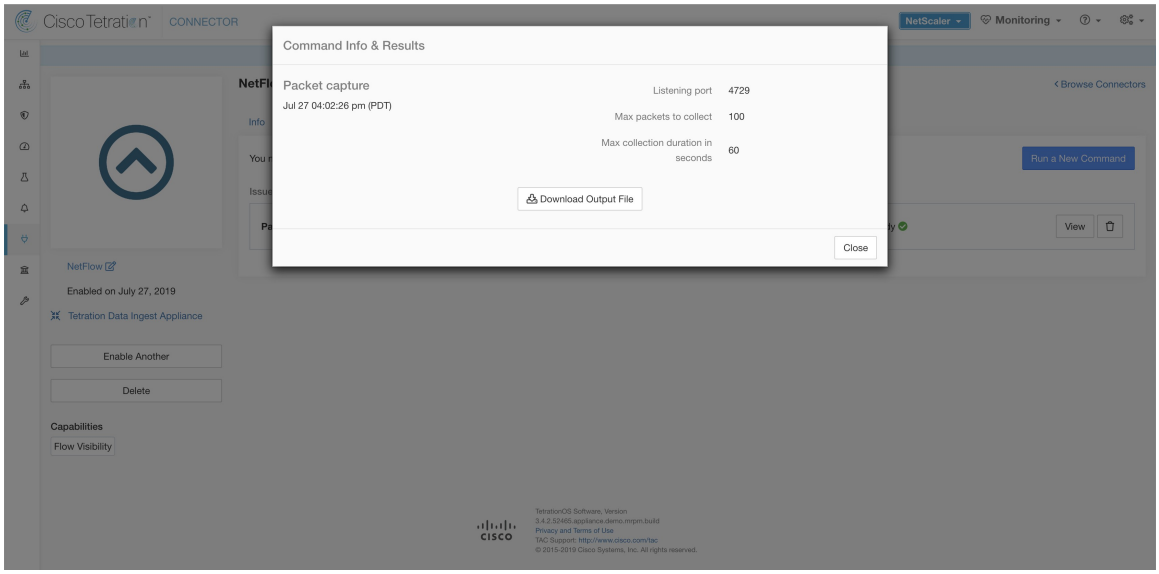
アプライアンスおよびコネクタで着信パケットをキャプチャします。Secure Workload はコマンドが発行されたアプライアンスおよびコネクタにコマンドを送信します。アプライアンスおよびコネクタサービスのコントローラは、パケットをキャプチャしてエンコードし、その結果を Cisco Secure Workload に返します。結果が Cisco Secure Workload で利用可能になると、ダウンロードボタンが表示され、ファイルを `.pcap` 形式でダウンロードできるようになります。

引数名	タイプ	説明
リスニングポート（Listening port）	number	このポートで送受信されるパケットをキャプチャします
収集する最大パケット数（Max packets to collect）	number	結果を返すまでに収集する最大パケット数。1000 未満にする必要があります
秒単位の最大収集期間（Max collection duration in seconds）	number	結果を返すまでの収集の最長時間。600秒未満にする必要があります。

許可されている Secure Workload 仮想アプライアンス：すべて

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、E メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *141 : NetFlow* コネクタの特定ポートでのパケットキャプチャ



コネクタのリスニングポートを更新する

Secure Workload Ingest アプライアンスのコネクタのリスニングポートを更新します。Secure Workloadは、コマンドが発行されたアプライアンスのアプライアンスコントローラにコマンドを送信します。コントローラは次のアクションを実行します。

- コネクタに対応する Docker サービスを停止します。
- サービスの現在実行中の設定を収集します。
- Docker サービスを削除します。
- 新しいポートを使用するようにサービスの実行設定を更新します。
- 新しい公開されたポートを使用して、削除されたコンテナで使用されていたものと同じ Docker イメージから新しいコンテナを開始します。また、先ほど削除されたコンテナに Dockerボリュームがマウントされていた場合、同じボリュームが新しいコンテナにマウントされます。
- コネクタの新しい IP バインディングを Cisco Secure Workload に返します。
- Cisco Secure Workload は、テキストボックスに結果を表示します。

引数名	タイプ	説明
Connector ID	string	リスニングポートを更新する必要があるコネクタのコネクタ ID

引数名	タイプ	説明
リスニングポートのラベル（**Listening port label**）	dropdown	更新されるポートのタイプ。
	NET-FLOW9	NetFlow v9 リスニングポート
	IPFIX	IPFIX リスニングポート
リスニングポート	string	コネクタの新しいポート

許可された **Cisco Secure Workload** 仮想アプライアンス：Secure Workload Ingest

許可されているコネクタ：なし

図 ***142 : Secure Workload Ingest*** アプライアンスで ***Meraki*** コネクタのリスニングポートを ***2055*** に更新します

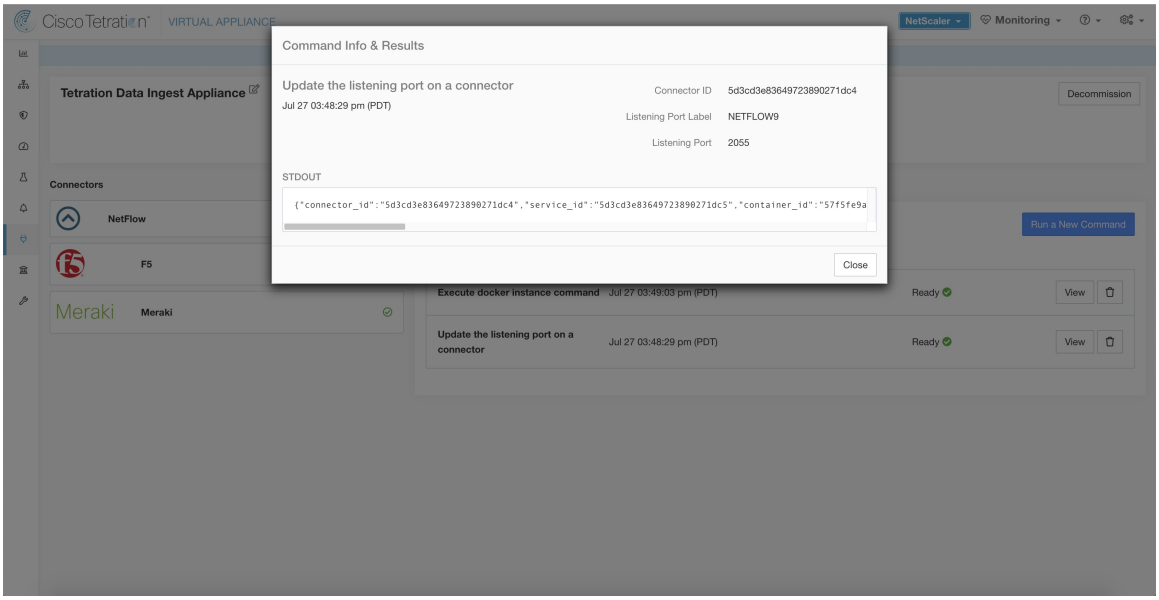
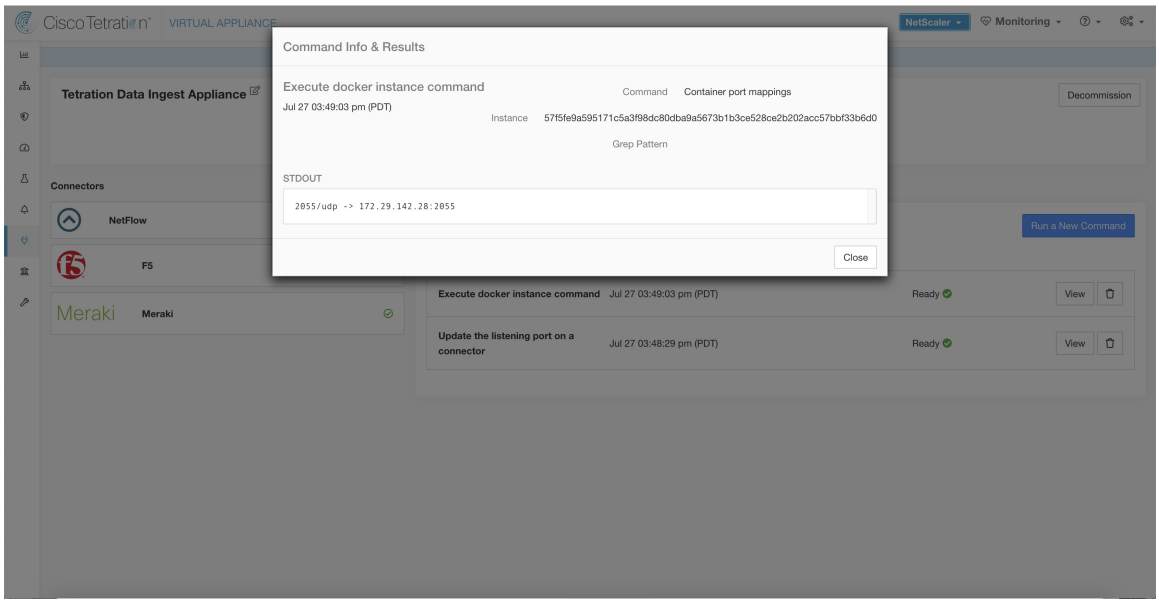


図 *143 : Secure Workload Ingest* アプライアンスで *Meraki* コネクタのポートマッピングを取得します



アラート通知コネクタログ構成の更新

Syslog、E メール、Slack、PagerDuty、および Kinesis アラート通知コネクタをホストする Secure Workload Alert Notifier（TAN）サービスのログ構成を更新します。TAN は複数のコネクタをホストしているため、ログ構成をコネクタページから直接更新することはできません。この許可されたコマンドにより、ユーザーはログ構成を更新できます。

Cisco Secure Workload は、Secure Workload Edge アプライアンス上の TAN Docker サービスのサービスコントローラにコマンドを送信します。コントローラはサービスに構成を適用し、構成更新のステータスを返します。

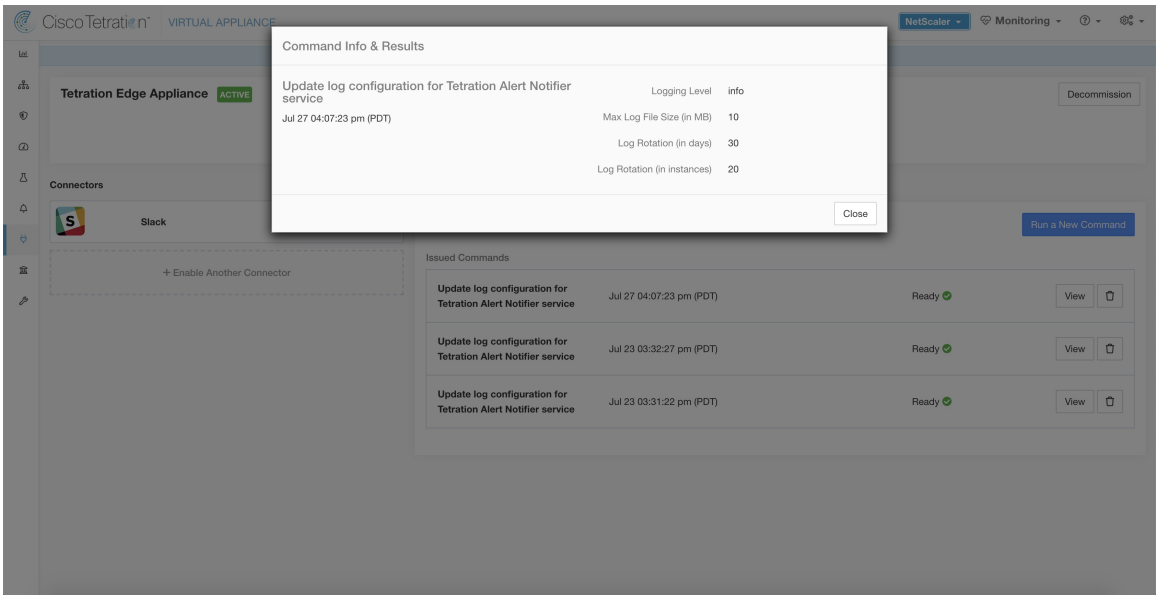
引数名	タイプ	説明
Logging level	dropdown	サービスで使用されるログレベル
	• *debug*	デバッグログレベル
	• *info*	情報ログレベル
	• *warn*	警告ログレベル
	• *error*	エラーログレベル
[最大ログファイルサイズ（MB 単位）（Max log file size (in MB)）]	number	ログ ローテーションが開始される前のログ ファイルの最大サイズ

引数名	タイプ	説明
[ログローテーション（日単位）（Log rotation (in days)）]	number	ログローテーションが開始されるまでのログ ファイルの最大経過時間
[ログローテーション（インスタンス単位）（Log rotation (in instances)）]	number	保持されるログファイルの最大インスタンス

許可されている Cisco Secure Workload 仮想アプライアンス：Cisco Secure Workload Edge

許可されているコネクタ：なし

図 *144 : Secure Workload Edge* アプライアンスの *Secure Workload Alert Notifier Docker* サービスのログ構成を更新する



アプライアンスからのスナップショットの収集

Cisco Secure Workload は、コマンドが発行されたアプライアンスにコマンドを送信します。アプライアンスのコントローラが Cisco Secure Workload からこのコマンドを受信すると、アプライアンスのスナップショットを収集してエンコードし、結果を Cisco Secure Workload に返します。結果が Cisco Secure Workload で利用可能になると、ダウンロードボタンが表示され、ファイルを `.tar.gz` 形式でダウンロードできるようになります。

スナップショットに含まれるファイル：

- /local/tetration/appliance/appliance.conf
- /local/tetration/{logs, sqlite, user.cfg}
- /opt/tetration/tet_vm_setup/conf/tet-vm-setup.conf
- /opt/tetration/tet_vm_setup/docker/Dockerfile

- /opt/tetration/ova/version
- /usr/local/tet-controller/conf
- /usr/local/tet-controller/cert/{topic.txt, kafkaBrokerIps.txt}
- /var/run/supervisord.pid

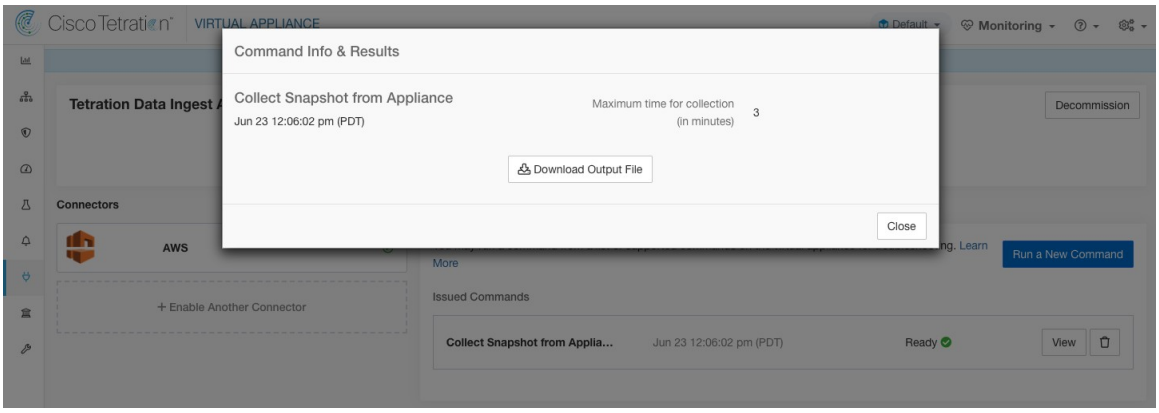
スナップショットに含まれるコマンド出力：

- ps aux
- iptables -L
- netstat {-nat, -rn, -suna, -stna, -tunlp}
- /usr/local/tet-controller/tet-controller -version
- supervisorctl status
- rpm -qi tet-nic-driver tet-controller
- du -shc /local/tetration/logs
- ls {/usr/local/tet-controller/cert/, -l /local/tetration/sqlite/, -l /opt/tetration/tet_vm_setup/.tet_vm.done, -l/opt/tetration/tet_vm_setup/templates/}
- docker {images, ps -a}
- blkid/ifconfig/lscpu/uptime
- free -m
- df -h

引数名	タイプ	説明
分単位での収集の最長時間（Max time for collection in minutes）	number	結果を返すまでの収集の最長時間。20 分未満である必要があります。

許可されている Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

図 *145 : Secure Workload* アプライアンスからのスナップショットの収集



コネクタからのスナップショットの収集

Cisco Secure Workload は、コネクタが展開されているアプライアンスにコマンドを送信します。コネクター ID に従って、コントローラはコネクタのスナップショットを収集し、それらをエンコードして、結果を Cisco Secure Workload に返します。結果が Cisco Secure Workload で利用可能になると、ダウンロードボタンが表示され、ファイルを .tar.gz 形式でダウンロードできるようになります。

スナップショットに含まれるファイル：

- /usr/local/tet-netflow/conf
- /local/tetration/{logs, sqlite}
- /var/run/{supervisord.pid, tet-netflow.pid}

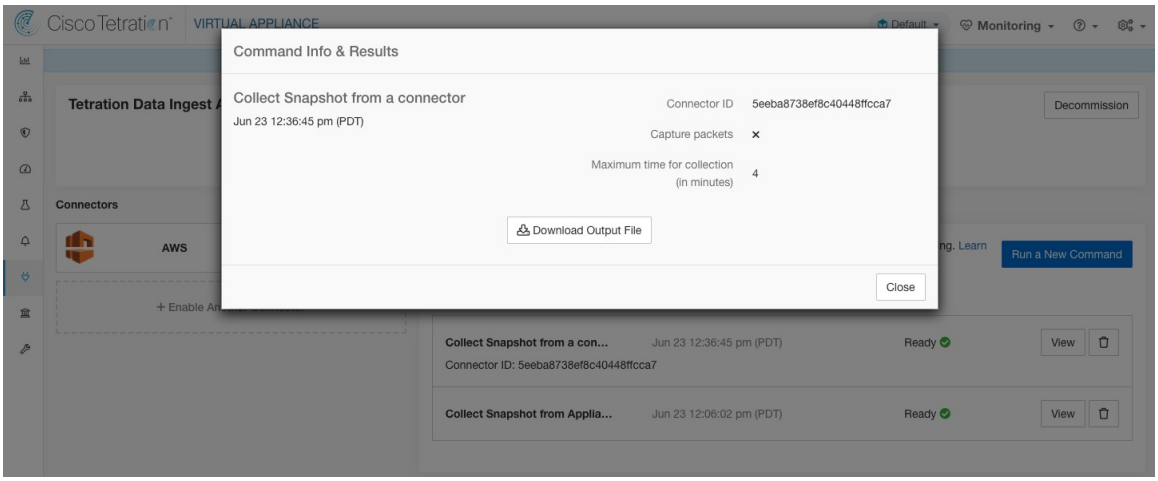
スナップショットに含まれるコマンド出力：

- ps aux
- netstat {-nat, -rn, -suna, -stna, -tunlp}

引数名	タイプ	説明
Connector ID	string	スナップショット コマンドが実行されるコネクタのコネクタ ID。
パケットのキャプチャ（Capture packets）	チェックボックス	パケットをキャプチャする必要があるかどうか。
収集の最大時間（Max time for collection in） **minutes**	number	結果を返すまでの収集の最長時間。20 分未満である必要があります。

許可されている Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

図 *146* : 指定されたコネクタ *ID* のコネクタからスナップショットを収集する *Secure Workload*



コントローラプロファイルの収集

アプライアンスまたはコネクタでのコントローラ プロセス プロファイリング結果を収集します。Secure Workloadは、コマンドが発行されたコネクタにコマンドを送信します。サービスコントローラは、指定されたプロファイリングモードでコネクタサービスを再起動します。プロファイリング結果を収集した後、サービスコントローラは、通常モードでサービスを再起動し、結果を Cisco Secure Workload に送信します。結果が Cisco Secure Workload で利用可能になると、ダウンロードボタンが表示され、ファイルを `.tar.gz` 形式でダウンロードできるようになります。

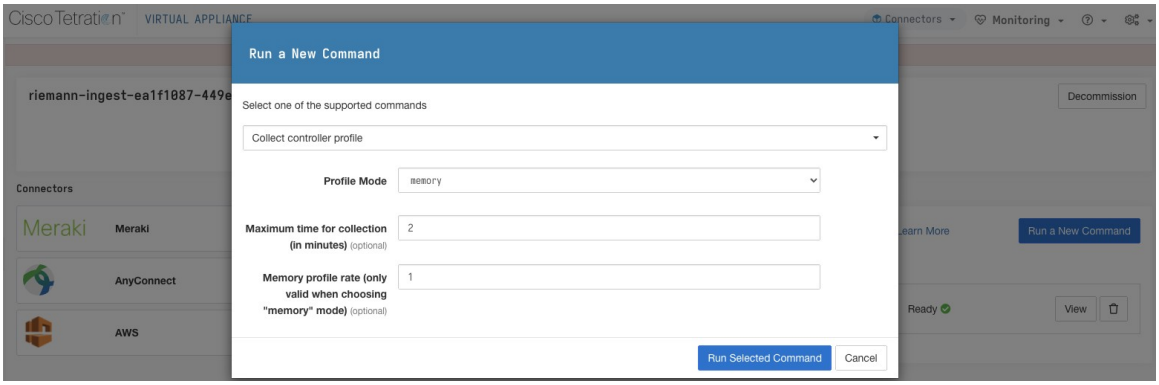
引数名	タイプ	説明
プロファイルモード	dropdown	プロファイリングモード。
	• *memory*	メモリ プロファイリング モード。
	• *cpu*	CPU プロファイリングモード。
	• *block*	ブロック プロファイリング モード。
	• *mutex*	Mutex プロファイリングモード。
	• *goroutine*	Goroutine プロファイリング モード。

引数名	タイプ	説明
収集の最長時間（分単位）	number	結果を返すまでの収集の最長時間。
メモリプロファイルレート（「メモリ」モードを選択した場合にのみ有効）	number	メモリ プロファイリング レート。このフィールドは任意です。指定しない場合、Golang のデフォルト値が使用されます。

許可されている Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、E メール、Slack、PagerDuty、Kinesis、ISE、Meraki。

図 *147 : Secure Workload* アプライアンスからのコントローラプロファイルの収集



コネクタプロファイルの収集

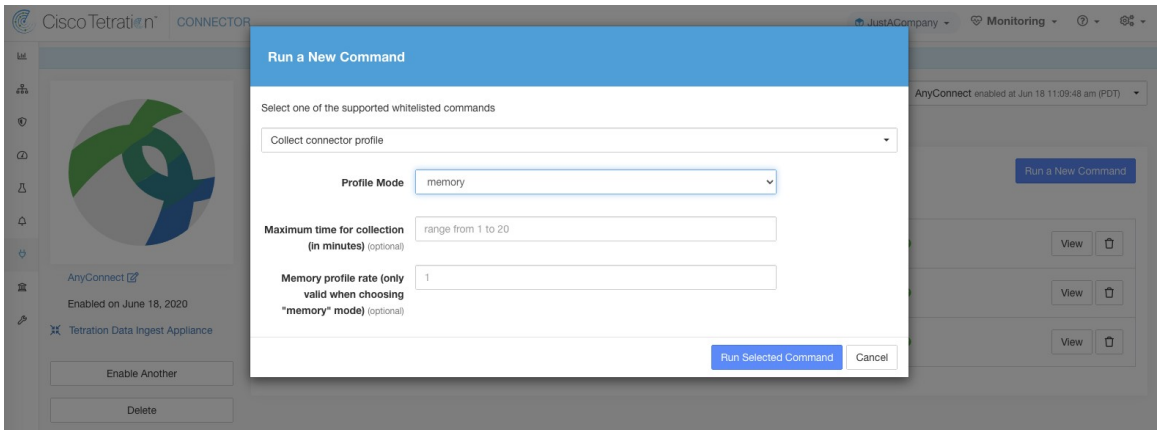
コネクタでのコネクタ プロセス プロファイリング結果を収集します。Secure Workload は、コマンドが発行されたコネクタにコマンドを送信します。サービスコントローラは、指定されたプロファイリングモードでコネクタサービスを再起動します。プロファイリング結果を収集した後、サービスコントローラは、通常モードでサービスを再起動し、結果を Cisco Secure Workload に送信します。結果が Cisco Secure Workload で利用可能になると、ダウンロードボタンが表示され、ファイルを `.tar.gz` 形式でダウンロードできるようになります。

引数名	タイプ	説明
プロファイルモード	dropdown	プロファイリングモード。
	• *memory*	メモリ プロファイリング モード。
	• *cpu*	CPU プロファイリングモード。
	• *block*	ブロック プロファイリング モード。
	• *mutex*	Mutex プロファイリングモード。
	• *goroutine*	Goroutine プロファイリング モード。
収集の最長時間（分単位）	number	結果を返すまでの収集の最長時間。
メモリプロファイルレート（「メモリ」モードを選択した場合にのみ有効）	number	メモリ プロファイリング レート。このフィールドは任意です。指定しない場合、Golang のデフォルト値が使用されます。

許可されている Cisco Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、E メール、Slack、PagerDuty、Kinesis、ISE、Meraki。

図 *148 : Secure Workload* コネクタからのコネクタプロファイルの収集



アプライアンスのコネクタアラート間隔のオーバーライド

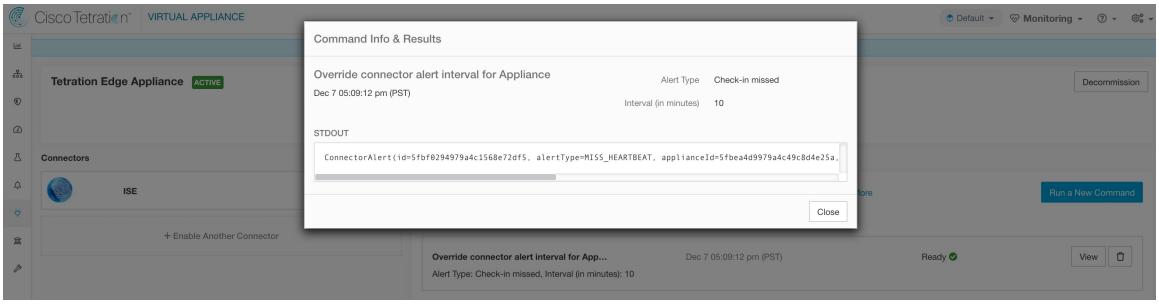
アプライアンスのデフォルトのコネクタアラート間隔をオーバーライドします。デフォルトで、Secure Workload では同じコネクタアラートが 1 日に 1 回だけ送信されるように制限されています。このコマンドは、管理者が 1 日 1 回の間隔では長すぎると考える場合に間隔をオーバーライドするためのものです。結果が Cisco Secure Workload で利用可能になると、テキストボックスに表示されます。

引数名	タイプ	説明
アラートタイプ	dropdown	オーバーライドするコネクタアラートタイプ。
	• チェックイン未実行（Check-in missed）	アプライアンスのチェックインが未実行。
	• *CPU* 使用率	高い CPU 使用率。
	• メモリ使用量	高いメモリ使用量。
	• ディスク使用量	高いディスク使用量。
[間隔（分単位）（Interval (in minutes)）]	number	間隔をオーバーライドする期間（分単位）。

許可されている Cisco Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

許可されているコネクタ：なし

図 *149 : Secure Workload* アプライアンスのコネクタアラート間隔のオーバーライド



コネクタのコネクタアラート間隔のオーバーライド

コネクタのデフォルトのコネクタアラート間隔をオーバーライドします。デフォルトで、Secure Workload では同じコネクタアラートが 1 日に 1 回だけ送信されるように制限されています。このコマンドは、管理者が 1 日 1 回の間隔では長すぎると考える場合に間隔をオーバーライドす

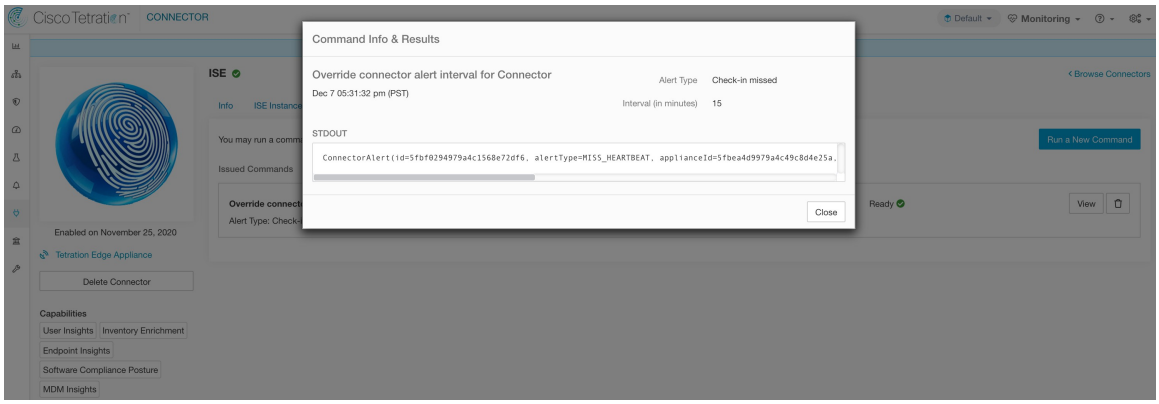
るためのものです。結果が Cisco Secure Workload で利用可能になると、テキストボックスに表示されます。

引数名	タイプ	説明
アラートタイプ	dropdown	オーバーライドするコネクタアラートタイプ。
	• チェックイン未実行（Check-in missed）	コネクタのチェックインが未実行です。
[間隔（分単位）（Interval (in minutes)）]	number	間隔をオーバーライドする期間（分単位）。

許可されている Secure Workload 仮想アプライアンス：なし

許可されているコネクタ：NetFlow、NetScaler、F5、AnyConnect、Syslog、電子メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki、ServiceNow、WAD。

図 *150 : Secure Workload* コネクタのコネクタアラート間隔のオーバーライド



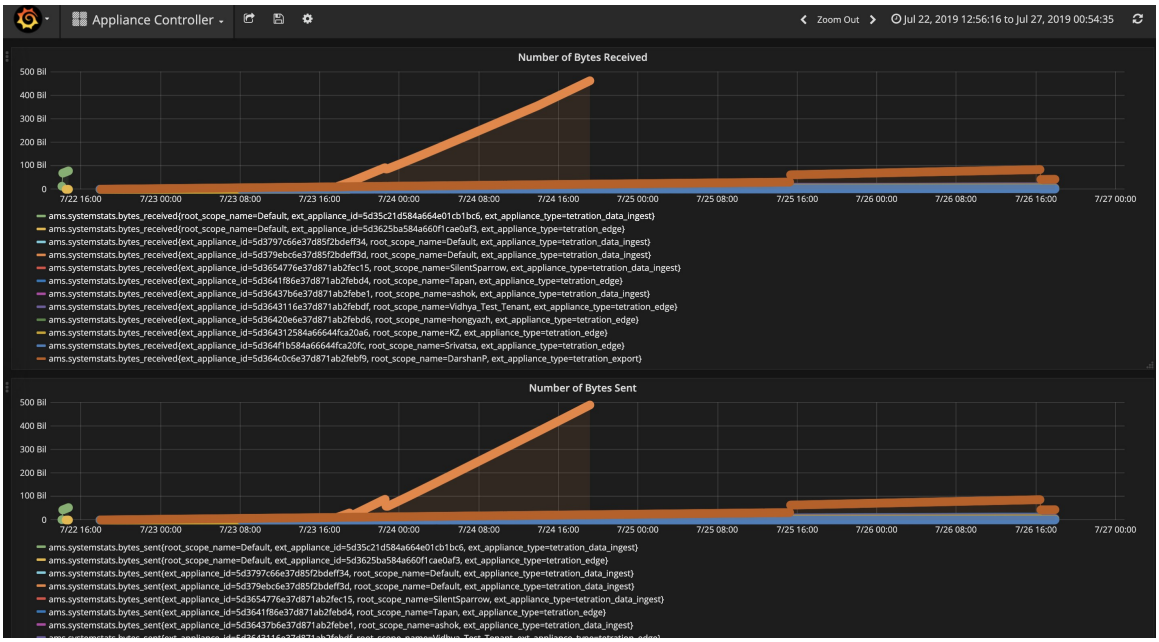
Hawkeye ダッシュボード

Hawkeye ダッシュボードには、コネクタ、およびコネクタが有効になっている仮想アプライアンスの正常性に関するインサイトが表示されます。

アプライアンスコントローラ ダッシュボード

アプライアンス コントローラ ダッシュボードは、ネットワーク統計に加えて、CPU 使用率、メモリ使用率、ディスク使用率、開いているファイル記述子の数などのシステムメトリックに関する情報を提供します。

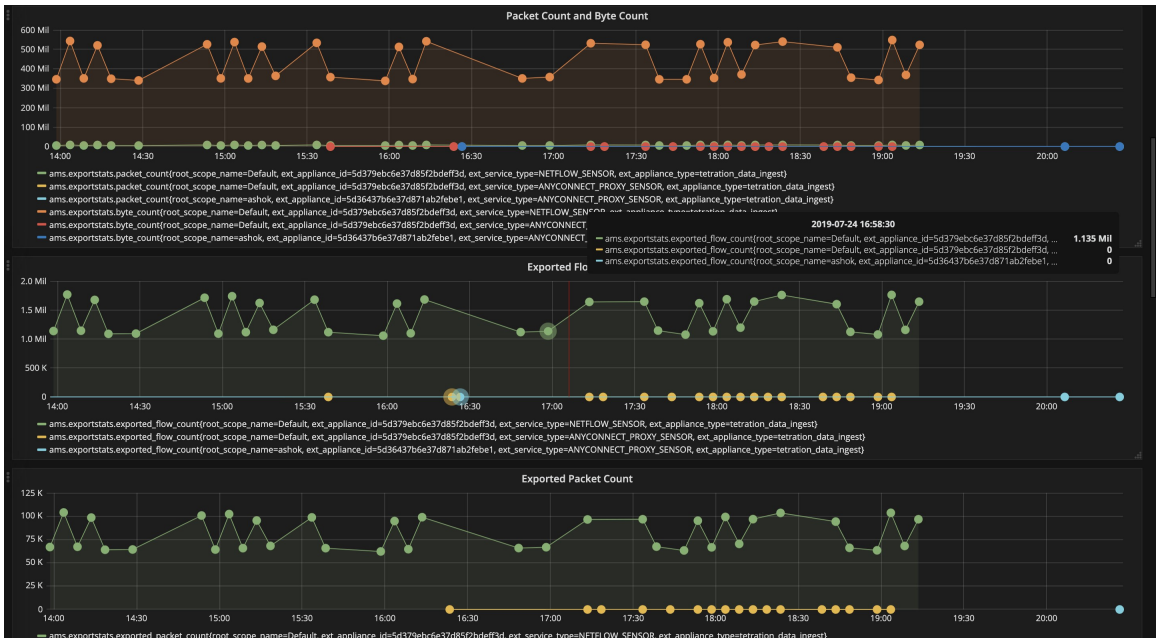
図 *151* : アプライアンス コントローラ ダッシュボード



サービスダッシュボード

サービスダッシュボードには、Cisco Secure Workload にエクスポートされたフロー観測数、Cisco Secure Workload にエクスポートされたパケット数、Cisco Secure Workload にエクスポートされたバイト数など、エクスポートメトリック（該当する場合）に関する情報が表示されます。さらに、このダッシュボードには、プロトコル処理と復号化に関する情報も表示されます（NetFlow v9 を処理するサービスや IPFIX など）。このダッシュボードでは、復号化の件数、復号化エラーの件数、フロー数、パケット数、バイト数などのメトリックを確認できます。さらに、サービスが実行されている Docker コンテナのシステムメトリックもこのダッシュボードに表示されます。CPU使用率、メモリ使用率、ディスク使用率、開いているファイル記述子の数などのメトリックは、このダッシュボードで提供されます。

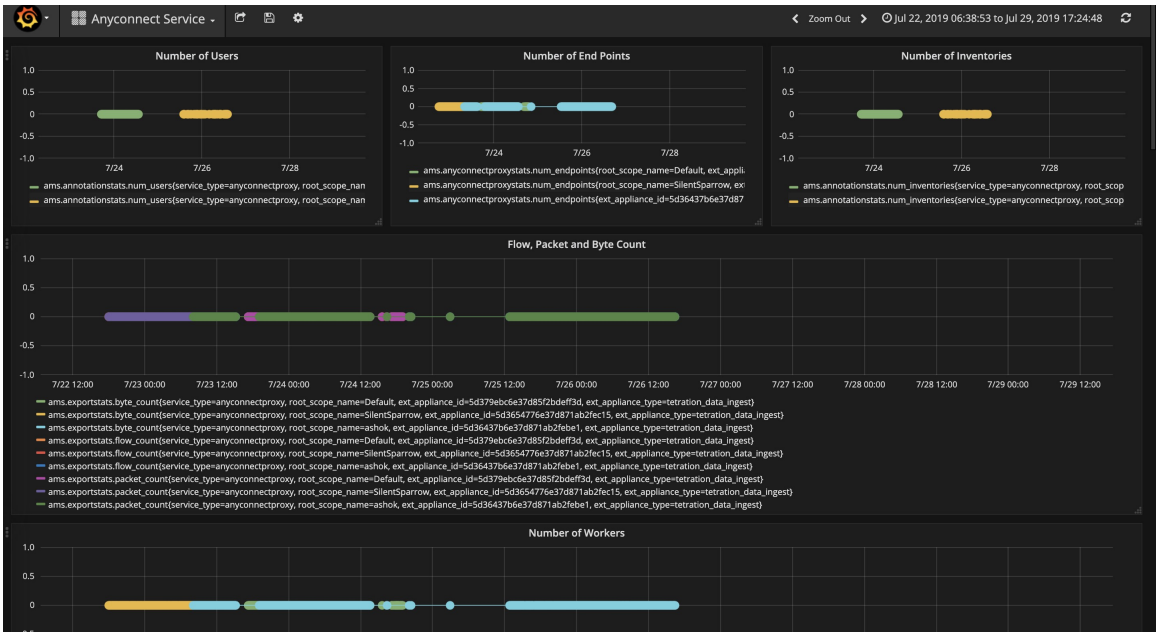
図 *152 :* サービスダッシュボード



AnyConnect サービスダッシュボード

AnyConnect サービスダッシュボードは、AnyConnect 固有のサービス情報に関する情報を提供します。このダッシュボードでは、エンドポイントの数、インベントリの数、ユーザーの数など、AnyConnect コネクタによって Secure Workload に報告されたメトリックを使用できます。さらに、このダッシュボードには、IPFIX プロトコルの処理とデコードに関する情報も表示されます。このダッシュボードでは、復号化されたカウント、復号化されたエラーカウント、フローカウント、パケットカウント、バイトカウントなどのメトリックを使用できます。

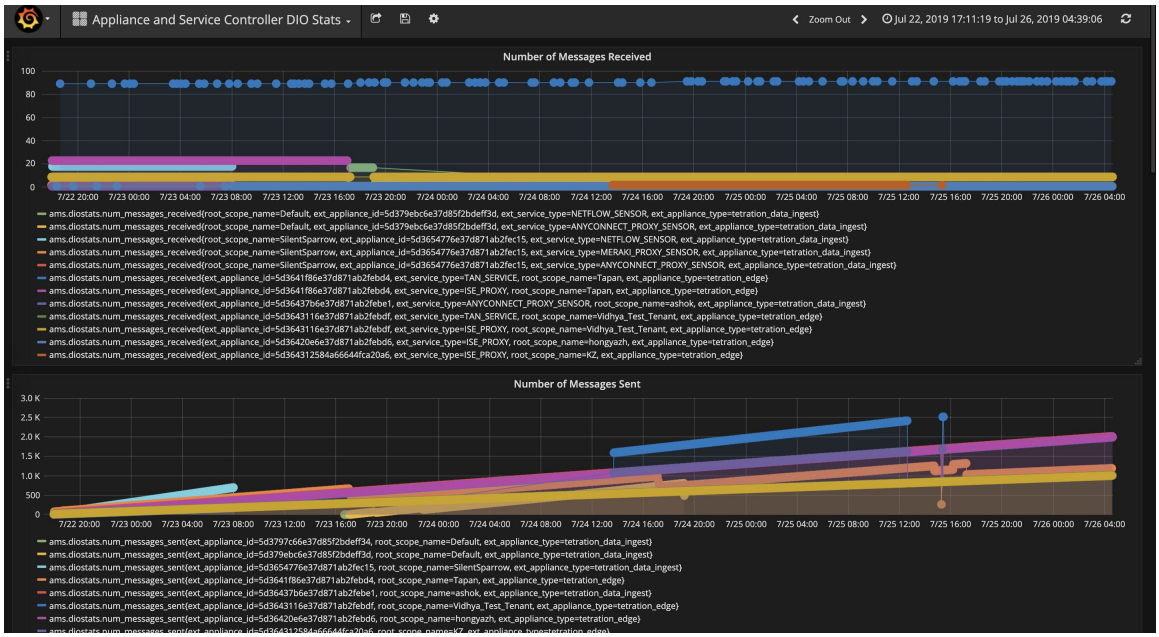
図 *153 : AnyConnect* ダッシュボード



アプライアンスとサービス DIO ダッシュボード

アプライアンスとサービスの DIO ダッシュボードは、アプライアンスマネージャとアプライアンス/サービスコントローラが通信する Kafka トピックで交換されたメッセージの数に関する情報を提供します。このダッシュボードには、受信したメッセージの数、送信したメッセージの数、失敗したメッセージの数などのメトリックが含まれています。さらに、コントローラによって読み取られた最後のオフセットが提供されるため、コントローラによるマネージャからの制御メッセージの処理が遅れているかどうかも把握できます。

図 *154* : アプライアンスとサービスの *DIO* ダッシュボード



一般的なトラブルシューティングのガイドライン

Cisco Secure Workload のコネクタページにコネクタがアクティブな状態で表示されたら、コネクタが有効になっているアプライアンスでアクションを実行する必要はありません。ユーザーはログインする必要はありません。この状態にならない場合は、次の情報がこのような問題のトラブルシューティングに役立ちます。

通常の状態では、アプライアンスで次のようになります。

- `systemctl status tet_vm_setup.service` は、*SUCCESS* 終了ステータスになっている *inactive* サービスをレポートします。
- `systemctl status tet-nic-driver` は、*active* サービスをレポートします。
- `Supervisorctl status tet-controller` は、*RUNNING* サービスをレポートします。これは、アプライアンスコントローラが稼働中であることを示します。
- `docker network ls` は、`bridge`、`host`、および `none` の 3 つのネットワークをレポートします。
- `docker ps` は、アプライアンスで実行されているコンテナをレポートします。通常、アプライアンスでコネクタが正常に有効化されると、アプライアンスで Docker コンテナがインスタンス化されます。Syslog、Email、Slack、PagerDuty、および Kinesis コネクタの場合、Secure Workload アラート通知サービスは Secure Workload Edge アプライアンスの Docker コンテナとしてインスタンス化されます。
- 各コンテナの `docker logs <cid>` は、tet-netflowsensor が RUNNING 状態になったことをレポートします。

- `docker exec <cid> ifconfig` は、ループバック以外に 1 つのインターフェイスのみをレポートします。
- `docker exec <cid> netstat -rn` は、デフォルトゲートウェイをレポートします。
- アプライアンスの `cat /local/tetration/appliance/appliance.conf` には、アプライアンスで実行されている Docker サービスのリストが表示されます。これには、サービス ID、コネクタ ID、コンテナ、イメージ ID、およびポートマッピング（該当する場合）に関する詳細が含まれます。Secure Workload Ingest アプライアンスでは、最大 3 つのサービスがアプライアンスで実行されています。ポートマッピングおよびコンテナにマウントされている Docker ボリュームが、このファイル内に存在します。

図 *155 : Secure Workload* アプライアンスの導入サービスとステータス

```
[root@esx-2106-ingest tetter]# systemctl status tet_vm_setup.service
● tet_vm_setup.service - Tetration Appliance Setup
   Loaded: loaded (/etc/systemd/system/tet_vm_setup.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since Sat 2019-07-27 23:51:29 UTC; 21h ago
 Main PID: 1249 (code=exited, status=0/SUCCESS)

Jul 27 23:51:12 localhost.localdomain python[1249]: mount: /dev/sr0 is write-protected, mounting read-only
Jul 27 23:51:29 esx-2106-ingest python[1249]: Docker version 18.09.8, build 0dd43dd87f
Jul 27 23:51:29 esx-2106-ingest python[1249]: REPOSITORY          TAG                 IMAGE ID            CREATE... SIZE
Jul 27 23:51:29 esx-2106-ingest python[1249]: userPrivateKey.key
Jul 27 23:51:29 esx-2106-ingest python[1249]: intermediateCA.cert
Jul 27 23:51:29 esx-2106-ingest python[1249]: kafkaBrokerIps.txt
Jul 27 23:51:29 esx-2106-ingest python[1249]: userCA.cert
Jul 27 23:51:29 esx-2106-ingest python[1249]: kafkaCA.cert
Jul 27 23:51:29 esx-2106-ingest python[1249]: topic.txt
Jul 27 23:51:29 esx-2106-ingest python[1249]: Created symlink from /etc/systemd/system/multi-user.target.wants/s...vice.
Hint: Some lines were ellipsized, use -l to show in full.
[root@esx-2106-ingest tetter]#
```

図 *156 : Secure Workload* ネットワーク ドライバ サービスのステータス

```
[root@esx-2106-ingest tetter]# systemctl status tet-nic-driver.service
● tet-nic-driver.service - NIC network driver plugin for Docker
   Loaded: loaded (/etc/systemd/system/tet-nic-driver.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2019-07-27 23:51:12 UTC; 21h ago
 Main PID: 733 (nic)
   Memory: 4.4M
   CGroup: /system.slice/tet-nic-driver.service
           └─733 /usr/local/tet/nic-driver/nic -log-level debug

Jul 27 23:51:12 localhost.localdomain systemd[1]: Started NIC network driver plugin for Docker.
Jul 27 23:51:12 localhost.localdomain systemd[1]: Starting NIC network driver plugin for Docker...
Jul 27 23:51:12 localhost.localdomain nic[733]: time="2019-07-27T23:51:12Z" level=info msg="NIC network driver started"
Hint: Some lines were ellipsized, use -l to show in full.
[root@esx-2106-ingest tetter]#
```

図 *157 :* アプライアンスコントローラのステータス

```
[root@esx-2106-ingest tetter]# supervisorctl status tet-controller
tet-controller                   RUNNING   pid 1971, uptime 21:43:29
[root@esx-2106-ingest tetter]#
```

上記のいずれにも当てはまらない場合は、/local/tetration/logs にある展開スクリプトのログを確認して、アプライアンスやコネクタの展開が失敗した理由を調べてください。

その他のコネクタ登録/接続の問題は、次のようにトラブルシューティングできます。

- `docker exec <cid> ps -ef` は、tet-netflowsensor-engine、`/usr/local/tet/ tet-netflowsensor -config /usr/local/tet-netflow/conf/tet-netflow.conf` インスタンス、およびプロセスマ

ネージャ /usr/bin/supervisord -c /usr/local/tet-netflow/ conf/supervisord.conf -n インスタンスをレポートします。

図 *158 : Secure Workload Injest* アプライアンスの *Cisco Secure Firewall ASA* コネクタでの実行中プロセス

```
[root@esx-2106-ingest tetter]# docker ps
CONTAINER ID        IMAGE                                                                                COMMAND
     CREATED             STATUS              PORTS                                NAMES
c82decfaa877        asa_sensor-3.4.2.52465.appliance.demo.mrpm.build-asa:5d3ce5e43649723890271dd3   "/usr/bin/supervisor
…"   22 hours ago        Up 22 hours         172.29.142.27:4729->4729/udp    asa-5d3ce5e43649723890271dd3
eddd5cd59839        aws_sensor-3.4.2.52465.appliance.demo.mrpm.build-aws:5d3ce3b73649723890271dce   "/usr/bin/supervisor
…"   22 hours ago        Up 22 hours                                              aws-5d3ce3b73649723890271dce
[root@esx-2106-ingest tetter]# docker exec c8 ps -ef
UID         PID  PPID  C STIME TTY          TIME CMD
root          1     0  0 00:01 ?        00:00:15 /usr/bin/python /usr/bin/supervisord -c /usr/local/tet-netflow/conf/supe
rvisord.conf -n
root          8     1  0 00:01 ?        00:02:24 /usr/local/tet-netflow/tet-netflowsensor-engine -ctrl-config /usr/local/
tet-netflow/conf/tet-controller.conf -upgrade-script /usr/local/tet-netflow/scripts/check_config_update.sh -service /usr
/local/tet-netflow/tet-netflowsensor -config /usr/local/tet-netflow/conf/tet-netflow.conf
root      27002     8  0 21:31 ?        00:00:00 /usr/local/tet-netflow/tet-netflowsensor -config /usr/local/tet-netflow/
conf/tet-netflow.conf
root      27024     0  0 21:32 ?        00:00:00 ps -ef
[root@esx-2106-ingest tetter]#
```

ログ ファイル

以下のコマンドを使用して、アプライアンスのさまざまなサービスからのログを表示できます。

- **/local/tetration/logs/tet-controller.log** は、アプライアンスコントローラのログを表示します。
- **docker exec <cid> cat /local/tetration/logs/tet-controller.log** は、コネクタのサービスコントローラのログを表示します。
- **docker exec <cid> cat /local/tetration/logs/tet-netflow.log** は、コネクタサービスのログを表示します。
- **docker exec <cid> cat /local/tetration/logs/tet-ldap-loader.log** は、LDAP スナップショット作成のログを表示します（LDAP 構成がコネクタに適用可能な場合）。
- **docker exec <cid> cat /local/tetration/logs/check_conf_update.log** は、構成更新のポーリングログを表示します（Ingest アプライアンスのコネクタの場合）。

（注） Secure Workload では、これらのログをアプライアンスやコネクタから直接プルできる一連のコマンドが許可されています。詳細については、「許可されている一連のコマンド」を参照してください

デバッグモード

アプライアンス/サービスコントローラとコネクタサービスのデフォルトのログレベルは、情報レベルに設定されています。問題をトラブルシューティングするには、エージェントをデバッグモードに設定する必要がある場合があります。これを行うには、対象のアプライアンス/コネクタについて、Secure Workload のアプライアンス/コネクタのログ設定を直接更新してく

ださい。コネクタの設定が更新されると、コントローラとサービスの両方のログレベルが更新されます。詳細については、「ログ設定」を参照してください。

コネクタアラート

アプライアンス/サービスに異常な動作が発生すると、コネクタアラートが作成されます。

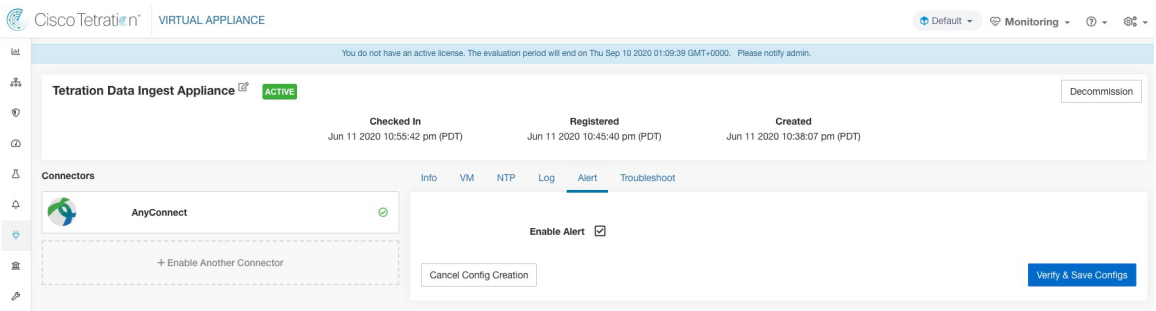
アラート設定

アプライアンスとコネクタのアラート設定により、ユーザーはさまざまなイベントに対してアラートを生成できます。3.4 リリースでは、この設定により、構成されているアプライアンスやコネクタで可能性のあるすべてのタイプのアラートが有効になります。

パラメータ名	タイプ	説明
Enable Alert	チェックボックス	アラートを有効にする必要があるかどうか。

（注） *Enable Alert* のデフォルト値は *true* です。

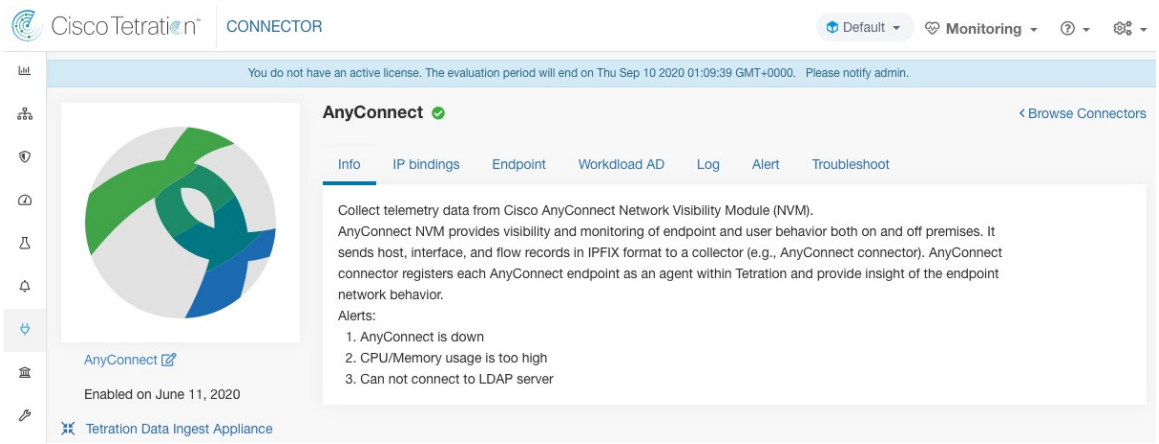
図 *159 : Secure Workload Data Ingest* アプライアンスでのアラート設定の表示



アラートタイプ

アプライアンスとコネクタごとに、アラートタイプは異なります。これは、アプライアンスとコネクタのページの [情報（Info）] タブで確認できます。

図 *160* : アラートリスト情報



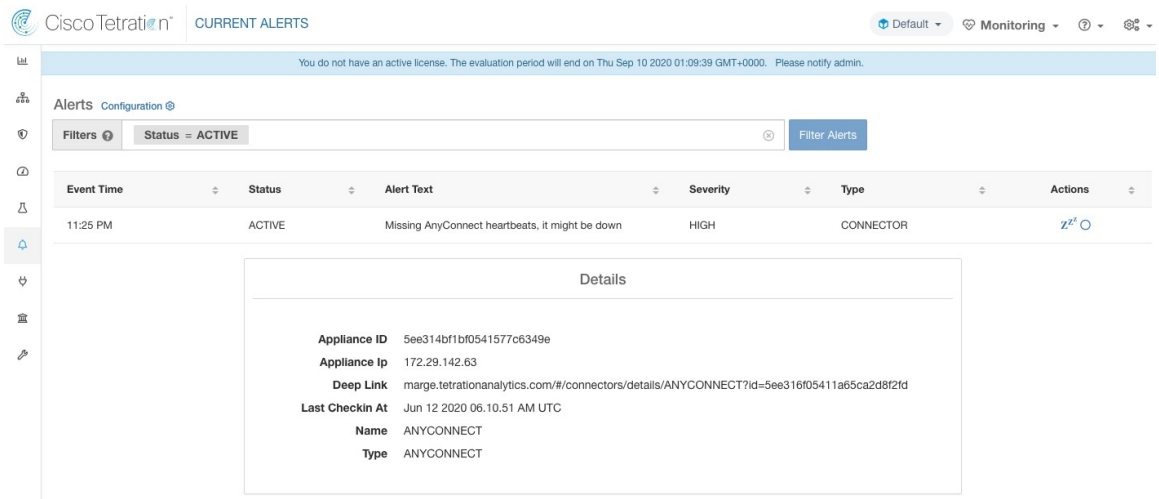
アプライアンス/コネクタダウン

このアラートは Cisco Secure Workload でアプライアンス/コネクタそれぞれからのハートビートが欠落しているために、アプライアンス（またはコネクタ）がダウンしている可能性がある場合に生成されます。

アラートテキスト：<アプライアンス/コネクタ> のハートビートがありません。ダウンしている可能性があります。（Missing <Appliance/Connector> heartbeats, it might be down.）

重大度：高

図 *161* : コネクタダウンのアラート



許可された Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

許可されたコネクタ：すべて

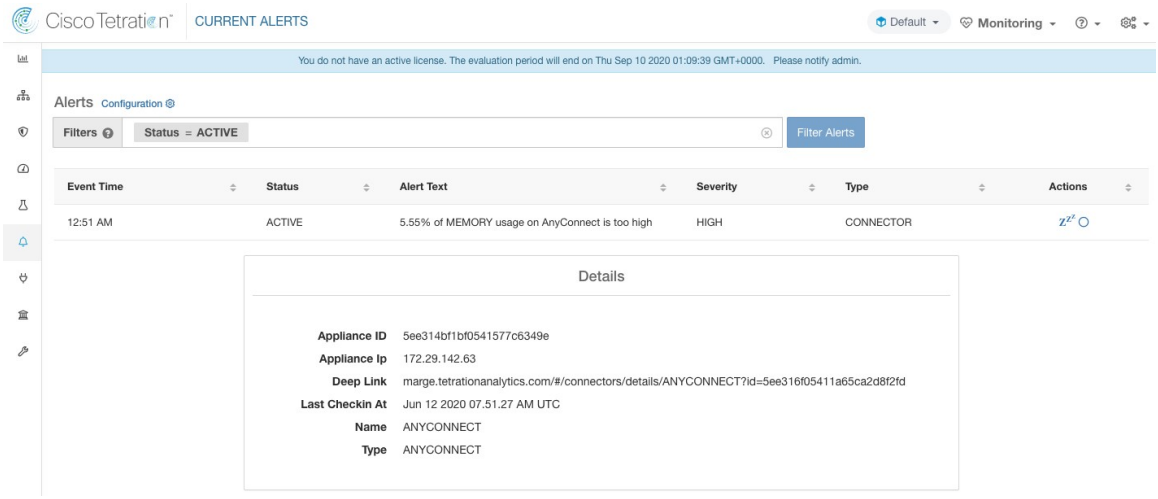
アプライアンスまたはコネクタシステムの使用率

アプライアンス（およびコネクタ）のシステム使用率（CPU、メモリ、およびディスク）が90%を超えると、この情報アラートが生成され、増加したシステム負荷をアプライアンス（および/またはコネクタ）が現在処理していることを示します。アプライアンスとコネクタが大量の処理アクティビティ中にシステムリソースの 90% 以上を消費することがありますが、これは正常です。

アラートテキスト：<アプライアンス/コネクタ> での <数字> 個の CPU/メモリ/ディスク使用率が高すぎます。

重大度：高

図 *162 :* コネクタシステムの使用率が高すぎる場合のアラート



許可された Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

許可されたコネクタ：すべて

コネクタ構成エラー

コネクタの構成から構成済みのサーバーに接続できない場合、このアラートは、承認されて展開された構成に潜在的な問題があることを示すために生成されます。たとえば、AnyConnect コネクタはLDAP構成を取得し、構成を検証して受け入れることができますが、通常の操作中に、構成が無効になる可能性があります。このアラートは、このシナリオをキャプチャし、構成を更新するための修正アクションをユーザーが実行する必要があることを示しています。

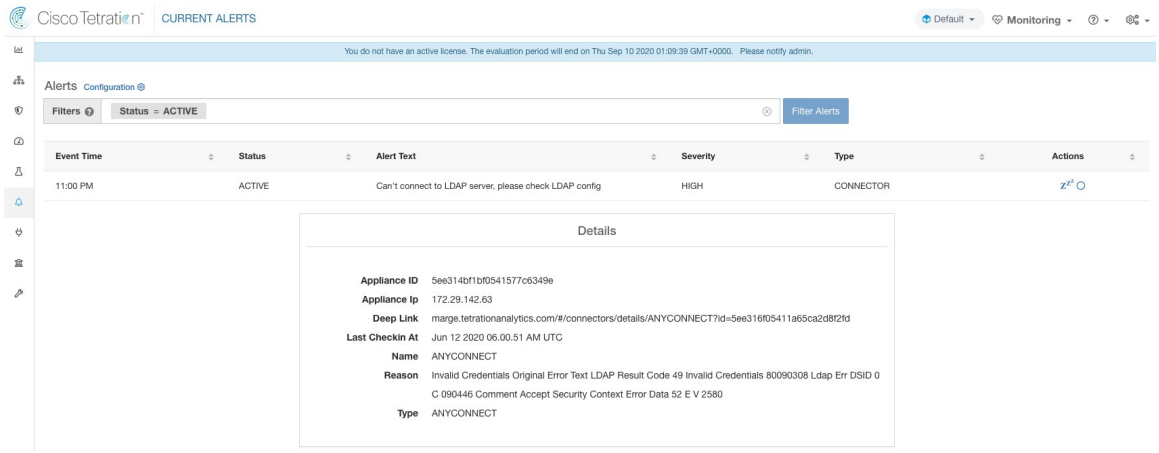
アラートテキスト：<Appliance/Connector> サーバーに接続できません。<Appliance/Connector> 構成を確認してください。

重大度：高、低

サーバー	コネクタ
LDAP サーバー	AnyConnect、F5、ISE、WDC

サーバー	コネクタ
ISE サーバー	ISE
ServiceNow サーバー	ServiceNow

図 *163* : 構成ステータスエラーのアラート

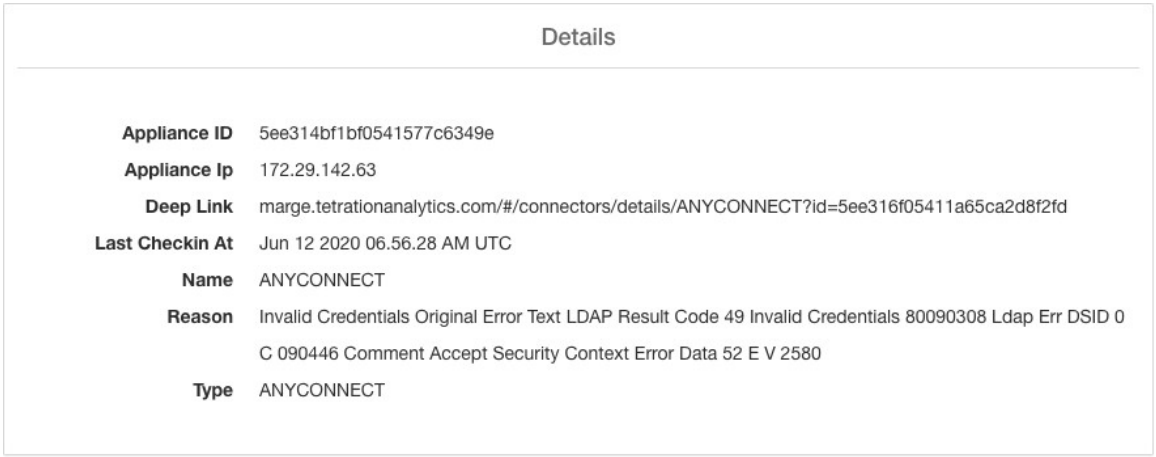


許可されている **Secure Workload** 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

許可されているコネクタ：AnyConnect、F5、ISE、WDC、ServiceNow。

コネクタ UI アラートの詳細

図 *164* : コネクタ *UI* アラートの詳細



アラート詳細

一般的なアラート構造とフィールドに関する情報については、「一般的なアラート構造」を参照してください。alert_details フィールドは構造化されており、コネクタアラートの次のサブフィールドが含まれます。

フィールド	タイプ	説明
アプライアンスID	文字列	アプライアンスID
アプライアンス IP	文字列	アプライアンス IP
Connector ID	文字列	Connector ID
コネクタ IP	文字列	コネクタ IP
ディープリンク	ハイパーリンク	アプライアンス/コネクタページにリダイレクト
最終チェックイン時間	文字列	最終チェックイン時間
Name	String	アプライアンス/コネクタ名
理由（Reason）	文字列	アプライアンス/コネクタがCisco Secure Workloadに接続できない理由
Type	文字列	アプライアンス/コネクタの種類

アラートの詳細の例

alert_details が json（文字列化されていない）として解析されると、次のようになります。

```
    {
      "Appliance ID": "5f1f3d26d674b01832c6792a",
      "Connector ID": "5f1f3e47baba512a70abee43",
      "Connector IP": "172.29.142.22",
      "Deep Link":
"bingo.tetrationanalytics.com/#/connectors/details/F5?id=5f1f3e47baba512a70abee43",
      "Last checkin at": "Aug 04 2020 20.37.33 PM UTC",
      "Name": "F5",
     "Reason": "Invalid Credentials (Original error text: LDAP Result Code 49 \"Invalid
 Credentials\": )",
      "Type": "F5"
    }
```

Cisco Secure Firewall Management Center

Secure Workload とシスコの Secure Firewall（旧称 Cisco Firepower）の能力を組み合わせることで、以下を利用したセキュリティソリューションを実現できます。

- [セグメンテーション（Segmentation）]

  ファイアウォールベースのセグメンテーションは、ソフトウェアエージェントがインストールされていないワークロードに適しています。ただし、この方法は、エージェントベースのワークロードにも使用できます。ネットワークに入るトラフィック、ネットワークから出るトラフィック、およびネットワーク内のワークロード間のトラフィックに対して、さまざまなポリシーセットを簡単かつ広く適用できます。

- 仮想パッチ適用

  仮想パッチ適用は、ソフトウェアエージェントがインストールされているワークロードに侵入防御システム（IPS）保護を追加します。この統合を使用して、悪意のあるトラフィックがアプリケーションに入ることを防ぎます。仮想パッチ適用設定により、Cisco Secure Workload は、IPS ポリシーの作成中に考慮すべき CVE を FMC に公開します。

この統合により、Cisco Secure Workload は、Cisco Secure Firewall Management Center インスタンスによって管理される Cisco Secure Firewall Threat Defense（旧称 Firepower Threat Defense）ファイアウォールでセグメンテーションポリシーを自動的に適用および管理します。ポリシーは動的に更新され、アプリケーション環境の変更に応じて、ポリシーが適用する一連のワークロードは継続的に更新されます。

ネットワークインベントリは、セグメンテーションポリシーのベースとなる Cisco Secure Workload インベントリフィルタによって動的に更新されます。ネットワークでワークロードが追加、変更、または削除されると、Cisco Secure Firewall Management Center 内のダイナミックオブジェクトが Cisco Secure Workload により自動的に更新されます。対応するアクセスコントロールルールは、これらのダイナミックオブジェクトに基づきます。適用されたポリシーの変更はすべて、管理対象の Cisco Secure Firewall Threat Defense（旧称 Firepower Threat Defense、FTD）デバイスに自動的に展開されます。Cisco Secure Firewall Management Center で変更を再展開する必要はありません。

動作の詳細、サポートされているプラットフォーム、制限、両方の製品のセットアップ手順、トラブルシューティング情報など、この統合に関する完全な情報については、「Cisco Secure Workload and Cisco Secure Firewall Management Center Integration Guide」を参照してください。

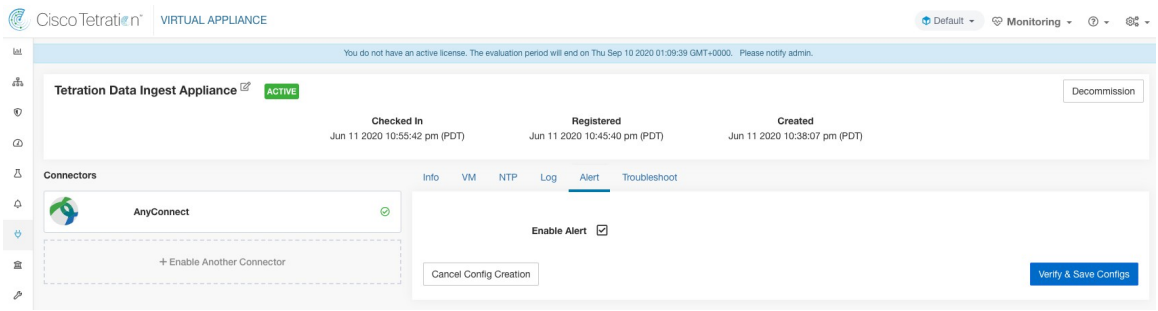
アラート設定

アプライアンスとコネクタのアラート設定により、ユーザーはさまざまなイベントに対してアラートを生成できます。3.4 リリースでは、この設定により、構成されているアプライアンスやコネクタで可能性のあるすべてのタイプのアラートが有効になります。

パラメータ名	タイプ	説明
Enable Alert	チェックボックス	アラートを有効にする必要があるかどうか。

（注） *Enable Alert* のデフォルト値は *true* です。

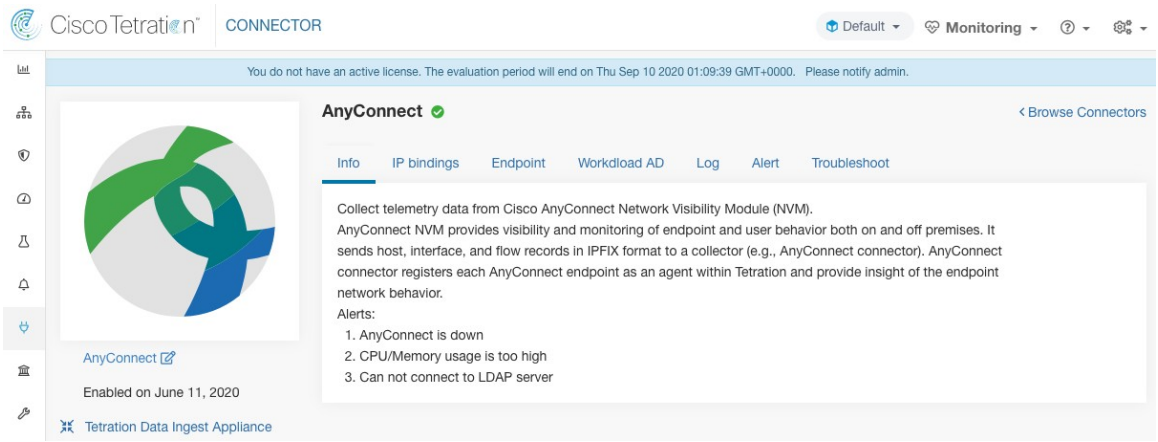
図 *165 : Secure Workload Data Ingest* アプライアンスでのアラート設定の表示



アラートタイプ

アプライアンスとコネクタごとに、アラートタイプは異なります。これは、アプライアンスとコネクタのページの [情報（Info）] タブで確認できます。

図 *166 :* アラートリスト情報



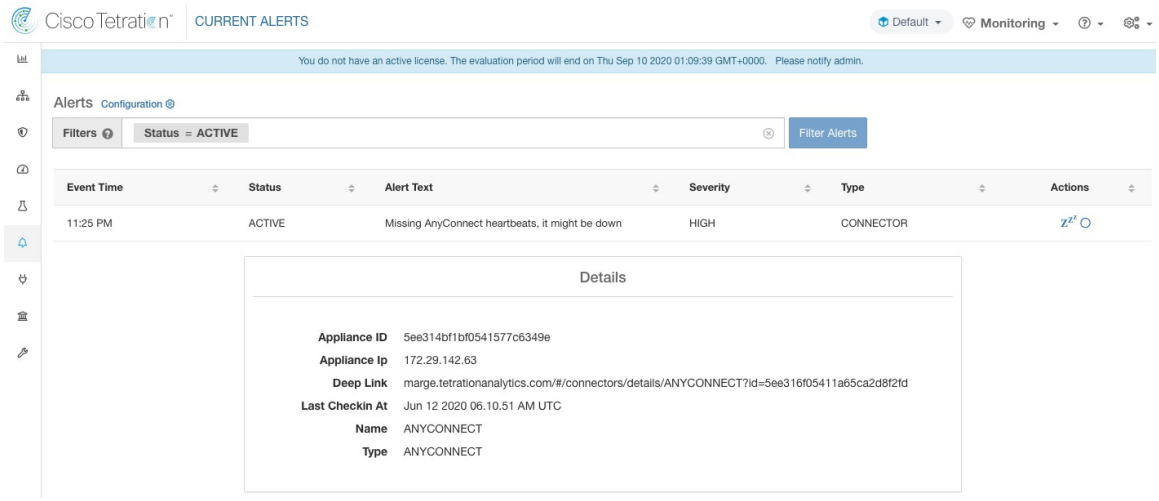
アプライアンス/コネクタダウン

このアラートは Cisco Secure Workload でアプライアンス/コネクタそれぞれからのハートビートが欠落しているために、アプライアンス（またはコネクタ）がダウンしている可能性がある場合に生成されます。

アラートテキスト：<アプライアンス/コネクタ> のハートビートがありません。ダウンしている可能性があります。（Missing <Appliance/Connector> heartbeats, it might be down.）

重大度：高

図 *167 :* コネクタダウンのアラート



許可された Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

許可されたコネクタ：すべて

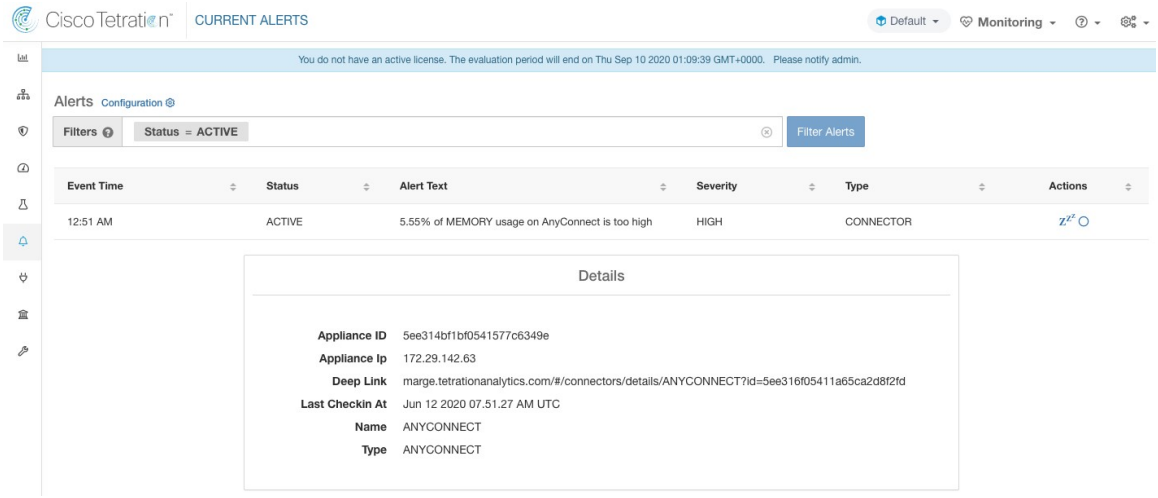
アプライアンスまたはコネクタシステムの使用率

アプライアンス（およびコネクタ）のシステム使用率（CPU、メモリ、およびディスク）が 90% を超えると、この情報アラートが生成され、増加したシステム負荷をアプライアンス（および/またはコネクタ）が現在処理していることを示します。アプライアンスとコネクタが大量の処理アクティビティ中にシステムリソースの 90% 以上を消費することがありますが、これは正常です。

アラートテキスト：<アプライアンス/コネクタ> での <数字> 個の CPU/メモリ/ディスク使用率が高すぎます。

重大度：高

図 *168* : コネクタシステムの使用率が高すぎる場合のアラート



許可された Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

許可されたコネクタ：すべて

コネクタ構成エラー

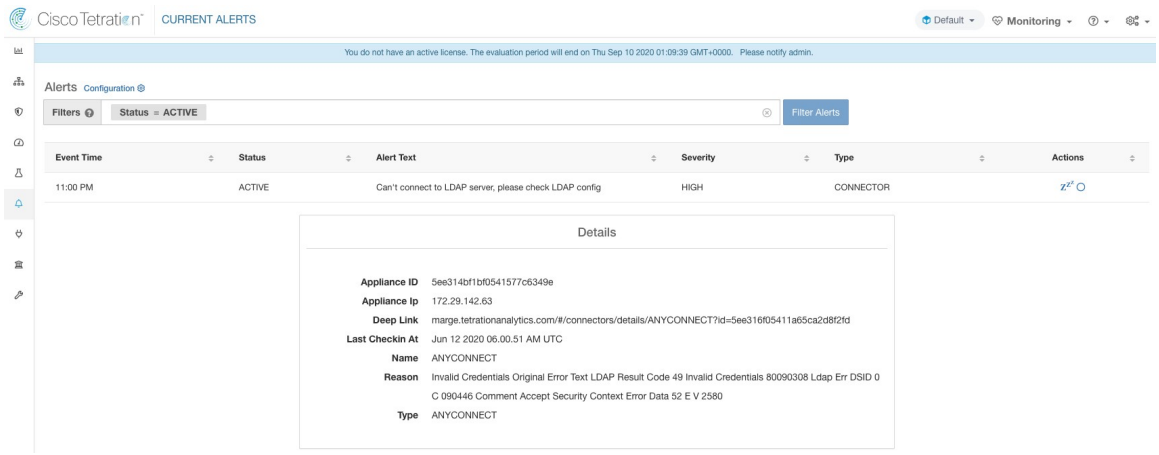
コネクタの構成から構成済みのサーバーに接続できない場合、このアラートは、承認されて展開された構成に潜在的な問題があることを示すために生成されます。たとえば、AnyConnect コネクタはLDAP構成を取得し、構成を検証して受け入れることができますが、通常の操作中に、構成が無効になる可能性があります。このアラートは、このシナリオをキャプチャし、構成を更新するための修正アクションをユーザーが実行する必要があることを示しています。

アラートテキスト：<Appliance/Connector> サーバーに接続できません。<Appliance/Connector> 構成を確認してください。

重大度：高、低

サーバー	コネクタ
LDAP サーバー	AnyConnect、F5、ISE、WDC
ISE サーバー	ISE
ServiceNow サーバー	ServiceNow

図 *169* : 構成ステータスエラーのアラート

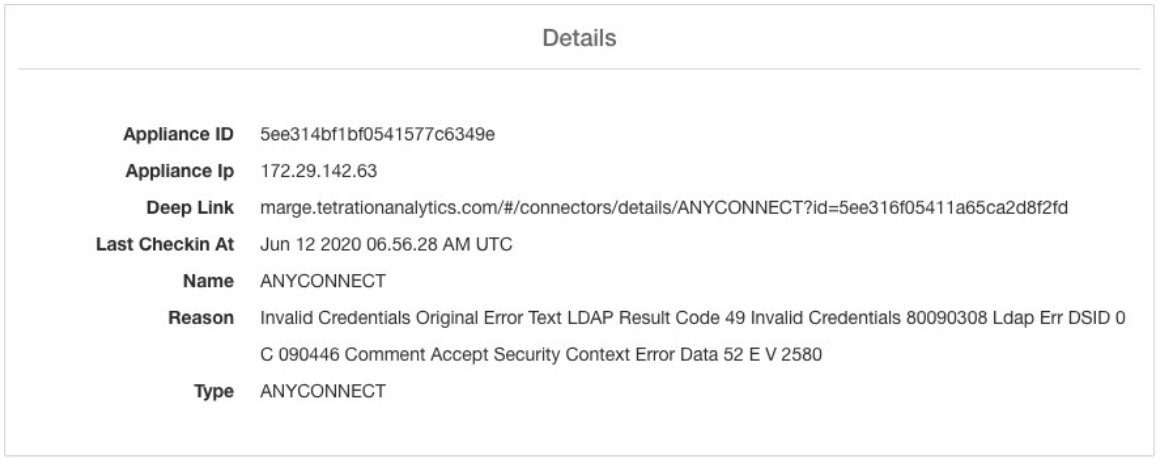


許可されている Secure Workload 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

許可されているコネクタ：AnyConnect、F5、ISE、WDC、ServiceNow。

コネクタ UI アラートの詳細

図 *170* : コネクタ *UI* アラートの詳細



アラート詳細

一般的なアラート構造とフィールドに関する情報については、「一般的なアラート構造」を参照してください。alert_details フィールドは構造化されており、コネクタアラートの次のサブフィールドが含まれます。

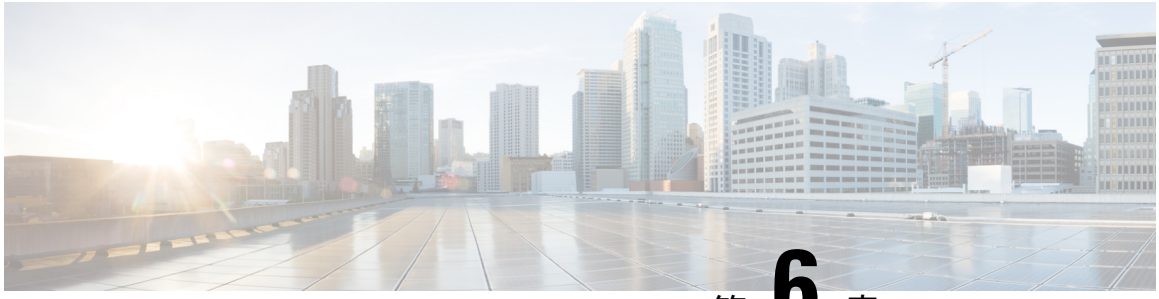
フィールド	タイプ	説明
アプライアンスID	文字列	アプライアンスID

フィールド	タイプ	説明
アプライアンス IP	文字列	アプライアンス IP
Connector ID	文字列	Connector ID
コネクタ IP	文字列	コネクタ IP
ディープリンク	ハイパーリンク	アプライアンス/コネクタページにリダイレクト
最終チェックイン時間	文字列	最終チェックイン時間
Name	String	アプライアンス/コネクタ名
理由（Reason）	文字列	アプライアンス/コネクタがCisco Secure Workloadに接続できない理由
Type	文字列	アプライアンス/コネクタの種類

アラートの詳細の例

alert_details が json（文字列化されていない）として解析されると、次のようになります。

```
    {
      "Appliance ID": "5f1f3d26d674b01832c6792a",
      "Connector ID": "5f1f3e47baba512a70abee43",
      "Connector IP": "172.29.142.22",
      "Deep Link":
"bingo.tetrationanalytics.com/#/connectors/details/F5?id=5f1f3e47baba512a70abee43",
      "Last checkin at": "Aug 04 2020 20.37.33 PM UTC",
      "Name": "F5",
     "Reason": "Invalid Credentials (Original error text: LDAP Result Code 49 \"Invalid
 Credentials\": )",
      "Type": "F5"
    }
```



第 6 章

インベントリ

インベントリは、ネットワーク上のすべてのワークロードの IP アドレスであり、それらを説明するラベルやその他のデータで注釈が付けられています。インベントリには、ベアメタルまたは仮想マシン、コンテナ、およびクラウドで実行されているワークロードが含まれます。パートナーネットワークで実行されているワークロードも含まれる場合があります（該当する場合）。

インベントリデータの収集は反復的なプロセスです。単一の IP アドレスのさまざまなソースからのデータをマージでき、新規および変更された IP アドレスを更新できます。通常、時間の経過とともに、インベントリの管理は徐々に動的になります。

各インベントリ項目に関連付けられているラベルと注釈に基づいて、検索、フィルタ、および範囲を使用してインベントリを操作し、グループ化します。ポリシーは、インベントリに定義したフィルタと範囲によって定義されたワークロードのグループに適用されます。

インベントリを操作するためのオプションは付与されているロールによって異なりますが、多くの場合検索、フィルタ、アップロードが含まれます。

- ワークロードラベル （408 ページ）
- 範囲とインベントリ （421 ページ）
- フィルタ （452 ページ）
- 範囲/フィルタ変更の影響を確認 （455 ページ）
- インベントリプロファイル （460 ページ）
- ワークロード プロファイル （462 ページ）
- Software Packages （473 ページ）
- 脆弱性データの可視化 （476 ページ）
- サービス プロファイル （483 ページ）
- ポッドプロファイル （484 ページ）
- 近隣 （485 ページ）

ワークロードラベル

ラベル（タグ、注釈、属性、メタデータ、またはコンテキストと呼ばれることもありますが、これらの用語は必ずしも完全に同義ではありません）は、Cisco Secure Workload の能力の鍵です。

人間が読めるラベルでは、機能、場所、その他の基準に関してワークロードについて説明します。

Cisco Secure Workload は、以下のユーザーラベルの追加方法をサポートしています。

- インベントリ項目で実行されている Secure Workload エージェントによる検出
- カンマ区切り値（CSV）ファイルのアップロードによる手動インポート
- ユーザーインターフェイスによる手動割り当て
- エンドポイントのコネクタによる自動インポート
- インベントリ強化用のコネクタによる自動インポート
- オーケストレータで生成されたラベルとカスタムラベルの自動インポート（「外部オーケストレータ」を参照）
- クラウドコネクタからの自動インポート（「Cloud Connector」を参照）
- インストーラスクリプトを作成するときに、インベントリラベルを指定できます。スクリプトを使用してインストールされたエージェントにはすべて、インベントリラベルが自動的にタグ付けされます。この機能は、Linux および Windows ワークロードの展開でのみサポートされています。

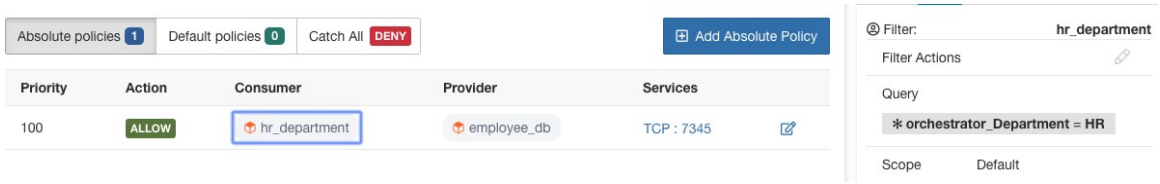
ラベルの重要性

ラベルを使用すると、論理ポリシーを定義できます。次に例を示します。

コンシューマ *hr_department* からプロバイダー *employee_db* へのトラフィックを許可する場合

コンシューマとプロバイダーのワークロード グループ メンバーを指定する代わりに、次の図に示すように、ラベルを使用して論理ポリシーを定義できます。これにより、論理ポリシーを変更することなく、コンシューマグループとプロバイダーグループのメンバーシップを動的に変更できるようになります。フリートからワークロードが追加または削除されるとと、外部オーケストレータやクラウドコネクタなど、設定したサービスによって Secure Workload に通知されます。これにより、Secure Workload はコンシューマグループ *hr_department* とプロバイダーグループ *employee_db* のメンバーシップを評価できます。

図 *171* : ラベル付きポリシーの例



サブネットベースのラベル継承

サブネットベースのラベル継承がサポートされます。下位のサブネットと IP アドレスは、次の条件のいずれかが満たされている場合、それらが属する上位のサブネットからラベルを継承します。

- 下位サブネット/アドレスのラベルのリストに該当するラベルが含まれていない。
- 下位サブネット/アドレスのラベル値が空である。

次の例を考えてみます。

IP	名前	目的	環境	スピリットアニマル
10.0.0.1	server-1	webtraffic	実稼働	
10.0.0.2				カエル
10.0.0.3				ワシ
10.0.0.0/24	web-vlan		統合	
10.0.0.0/16		webtraffic		アナグマ
10.0.0.0/8			test	クマ

IP アドレス *10.0.0.3* のラベルは *{“name”: “web-vlan”, “purpose”: “webtraffic”, “environment”: “integration”, “spirit-animal”: “eagle”}* です。

ラベルのプレフィックス

ラベルは、情報のソースを識別するプレフィックス付きで自動的に表示されます。

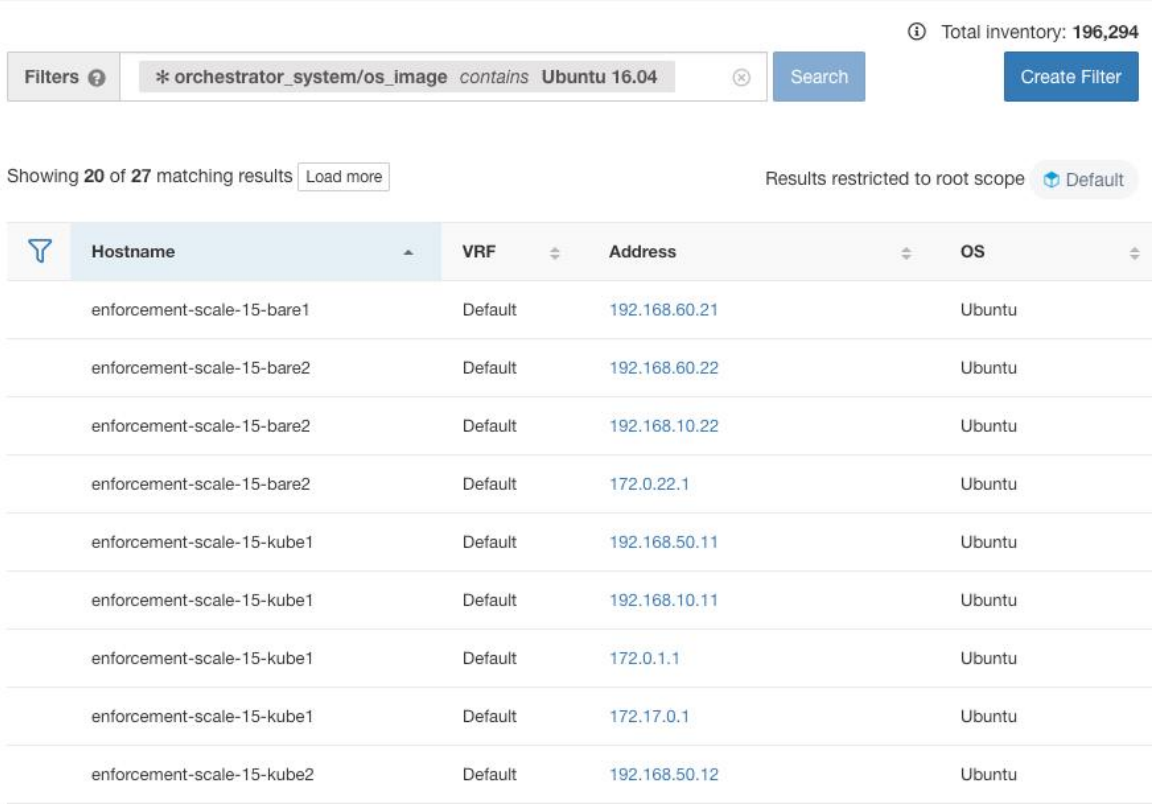
UI（*OpenAPI* では *user_*）では、すべてのユーザーラベルの先頭に * が付きます。さらに、外部オーケストレータまたはクラウドコネクタから自動的にインポートされたラベルには、orchestrator_ というプレフィックスが付きます。エンドポイントコネクタからインポートされたラベルについては、「エンドポイントのコネクタ」で詳細を参照してください。ただし、*ldap_* のプレフィックスが付いたラベルが含まれる場合があります。

たとえば、ユーザーがアップロードしたCSVファイルからインポートされた*department*のキーを持つラベルは、UIでは** department*と表示され、OpenAPIでは*user_department*と表示されます。外部オーケストレータからインポートされた*location*のキーを持つラベルは、UIでは** orchestrator_location*と表示され、OpenAPIでは*user_orchestrator_location*と表示されます。

次の図は、オーケストレータが生成したプレフィックス付きのラベルを使用したインベントリ検索の例を示しています。

orchestrator_system/os_image:

図 *172*：オーケストレータで生成されたラベルを使用したインベントリ検索の例



クラウドコネクタによって生成されたラベル

これらのラベルは、AWSおよびAzureからのデータに適用されます。これらのラベルの送信元は、AWS VPCまたはAzure VNetのワークロードとネットワーク インターフェイスです。送信元からのタグがマージされ、Cisco Secure Workloadに表示されます。たとえば、ワークロードタグが`env: prod`で、ネットワーク インターフェイス タグが`env: test`である場合、Cisco Secure Workloadのラベル値は`prod,test`であり、各コネクタページの[orchestrator_env]列の下に表示されます。

AKS、EKS、GKEに固有のラベルについては、「Kubernetesクラスタに関連するラベル」も参照してください。

表 *22*：クラウドコネクタを使用して収集されたすべてのインベントリに追加されるラベル

キー	値
orchestrator_system/orch_type	AWS または Azure
orchestrator_system/cluster_name	<*Cluster_name* はユーザーがこのコネクタの設定に付けた名前>
orchestrator_system/cluster_id	<仮想ネットワーク *ID*>

インスタンス固有のラベル

次のラベルは、各ノードに固有のものです。

キー	値
orchestrator_system/workload_type	vm
orchestrator_system/machine_id	<プラットフォームによって割り当てられた *InstanceID*>
orchestrator_system/machine_name	<AWS によってこのノードに指定された PublicDNS（FQDN）> – または – <Azure での InstanceName>
orchestrator_system/segmentation_enabled	<インベントリでセグメンテーションが有効になっているかどうかを判断するためのフラグ>
orchestrator_system/virtual_network_id	<インベントリが属する仮想ネットワークの *ID*>
orchestrator_system/virtual_network_name	<インベントリが属する仮想ネットワークの名前>
orchestrator_system/interface_id	<このインベントリに付属する elastic network interface の識別子>
orchestrator_system/region	<インベントリが属する地域>
orchestrator_system/resource_group	（このタグは Azure インベントリにのみ適用されます）
orchestrator_‘<Tag Key>‘	<Tag Value> クラウドポータルのインベントリに割り当てられた任意数のカスタムタグのキーと値のペア。

Kubernetes クラスタに関連するラベル

次の情報は、シンプルな Kubernetes、OpenShift、およびサポートされているクラウドプラットフォーム（EKS、AKS、および GKE）で実行されている Kubernetes に適用されます。

Secure Workload は、オブジェクトタイプごとに、オブジェクトに関連付けられたラベルを含むインベントリを Kubernetes クラスタからリアルタイムでインポートします。ラベルのキーと値はそのままインポートされます。

Secure Workload では、Kubernetes オブジェクト用に定義されたラベルのインポートに加えて、これらのオブジェクトをインベントリフィルタで使用しやすくするラベルも生成します。これらの追加のラベルは、範囲とポリシーを定義する際に特に役立ちます。

すべてのリソースに対して生成されたラベル

Secure Workload は、Kubernetes/OpenShift/EKS/AKS/GKE API サーバーから取得したすべてのノード、ポッド、およびサービスに次のラベルを追加します。

キー	値
orchestrator_system/orch_type	kubernetes
orchestrator_system/cluster_id	<*UUID of the cluster's configuration in |product|*>
orchestrator_system/cluster_name	<*Name given to this cluster's configuration*>
orchestrator_system/namespace	<この項目の *Kubernetes/OpenShift/EKS/AKS/GKE* 名前空間>

ノード固有のラベル

次のラベルは、ノードに対してのみ生成されます。

キー	値
orchestrator_system/workload_type	マシン
orchestrator_system/machine_id	<*Kubernetes/OpenShift* によって割り当てられた *UUID*>
orchestrator_system/machine_name	<このノードに指定された名前>
orchestrator_system/kubelet_version	<このノードで実行されている *kubelet* のバージョン>
orchestrator_system/container_runtime_version	<このノードで実行されているコンテナ ランタイム バージョン>

ポッド固有のラベル

次のラベルは、ポッドに対してのみ生成されます。

キー	値
orchestrator_system/workload_type	pod
orchestrator_system/pod_id	<*Kubernetes/OpenShift* によって割り当てられた *UUID*>
orchestrator_system/pod_name	<このポッドに指定された名前>

キー	値
orchestrator_system/hostnetwork	<*true*|*false*> ポッドがホストネットワークで実行されているかどうかを反映
orchestrator_system/machine_name	<ポッドが実行されているノードの名前>
orchestrator_system/service_endpoint	*[この*Pod*が提供しているサービス名のリスト]*

サービス固有のラベル

次のラベルは、サービスに対してのみ生成されます。

キー	値
orchestrator_system/workload_type	サービス
orchestrator_system/service_name	<このサービスに指定された名前>

- （クラウドマネージド型 Kubernetes の場合のみ）ServiceType：LoadBalancer のサービスは、メタデータの収集に対してのみサポートされ、フローデータの収集やポリシーの適用に対してはサポートされません。

ヒント **orchestrator_system/service_name** を使用して項目をフィルタリングすることと、**orchestrator_system/service_endpoint** を使用することは同じではありません。

たとえば、フィルタ **orchestrator_system/service_name = web** を使用すると、**web** という名前のすべてのサービスが選択されますが、 **orchestrator_system/service_endpoint = web** は、**web** という名前のサービスを提供するすべてのポッドを選択します。

Kubernetes クラスタのラベルの例

次の例は、Kubernetes ノードの部分的な YAML 表現と、Cisco Secure Workload によってインポートされた対応するラベルを示しています。

```
- apiVersion: v1
kind: Node
metadata:
  annotations:
    node.alpha.kubernetes.io/ttl: "0"
    volumes.kubernetes.io/controller-managed-attach-detach: "true"
  labels:
    beta.kubernetes.io/arch: amd64
    beta.kubernetes.io/os: linux
    kubernetes.io/hostname: k8s-controller
```

表 *23 : Kubernetes* からインポートされたラベルキー

インポートされたラベルキー
orchestrator_beta.kubernetes.io/arch
orchestrator_beta.kubernetes.io/os
orchestrator_kubernetes.io/hostname
orchestrator_annotation/node.alpha.kubernetes.io/ttl
orchestrator_annotation/volumes.kubernetes.io/controller-managed-attach-detach
orchestrator_system/orch_type
orchestrator_system/cluster_id
orchestrator_system/cluster_name
orchestrator_system/namespace
orchestrator_system/workload_type
orchestrator_system/machine_id
orchestrator_system/machine_name
orchestrator_system/kubelet_version
orchestrator_system/container_runtime_version

カスタムラベルのインポート

カスタムラベルは、ユーザー定義データを特定のホストに関連付けるため、アップロードするか、手動による割り当てができます。このユーザー定義データは、関連するフローとインベントリに注釈を付けるために使用されます。

ラベルの入手元（手動入力または手動アップロード、コネクタまたは外部オーケストレータを使用した取り込みなど）にかかわらず、すべてのルート範囲でラベル付けできる IPv4/IPv6 アドレスまたはサブネットの数には制限があることに注意してください。詳細については、「ラベルの制限」を参照してください。

ラベルファイルのアップロードに関するガイドライン

手順

ステップ 1 サンプルファイルを表示するには、左側のウィンドウで**[整理（Organize）]**>**[ユーザーがアップロードしたラベル（User Uploaded Labels）]** > **[CSVアップロード（CSV Upload）]**を選択し、[サンプルのダウンロード（Download a Sample）] をクリックします。

ステップ 2 サンプルファイルを表示するには、左側のペインで **[整理（Organize）]** > **[ラベル管理（Label Management）]** > **[ユーザー定義ラベルのアップロード（User Defined Label Upload）]** を選択し、[サンプルのダウンロード（Download a Sample）] をクリックします。

ステップ 3 ユーザーラベルのアップロードに使用する CSV ファイルには、ラベルキー（IP アドレス）が含まれている必要があります。

ステップ 4 ラベルに英語以外の文字を使用するには、CSV ファイルを UTF-8 形式にする必要があります。

ステップ 5 CSV ファイルが「ラベルキースキーマ」セクションで説明されているガイドラインを満たしていることを確認します。

ステップ 6 アップロードされたすべてのファイルは、同じスキーマに従う必要があります。

ラベルキースキーマ

列名に関するガイドライン

- ラベルキースキーマでは、ヘッダー「IP」を持つ 1 つの列が必要です。さらに、IP アドレスの属性を含む他の列が少なくとも 1 つ必要です。
- [VRF] 列は、ラベルスキーマで特別な意味を持ちます。この値を指定する場合は、ラベルがアップロードされるルート範囲と一致する必要があります。範囲に依存しない API を使用して CSV ファイルをアップロードする場合、これは必須です。
- 列名に使用できる文字は、文字、数字、スペース、ハイフン、アンダースコア、およびスラッシュのみです。
- 列名は 200 文字を超えることはできません。
- 列名の前に「orchestrator_」、「TA_」、「ISE_」、「SNOW_」、または「LDAP_」のプレフィックスを付けることはできません。これらは内部アプリケーションのラベルと競合する可能性があるためです。
- CSV ファイルに重複する列名が含まれないようにしてください。

列の値に関するガイドライン

- 値は 255 文字に制限されていますが、可能な限り短くする必要があると同時に、ユーザーにとって明確で、わかりやすく意味のあるものでなければなりません。
- キーと値は大文字と小文字が区別されません。ただし、一貫性を持たせることをお勧めします。
- 「IP」列の下に表示されるアドレスは、次の形式に従う必要があります。
  - IPv4 アドレスの形式は、「x.x.x.x」および「x.x.x.x/32」です。
  - IPv4 サブネットは、「x.x.x.x/<netmask>」の形式にする必要があります。ここで、ネットマスクは 0 から 31 までの整数です。

- IPv6 アドレスでは、長い形式（「x:x:x:x:x:x:x:x」または「x:x:x:x:x:x:x:x/128」）および標準形式（「x:x::x」または「x:x::x/128」）がサポートされています。
- IPv6 サブネットでは、長い形式（「x:x:x:x:x:x:x:x/<netmask>」）と標準形式（「x:x::x/<netmask>」）がサポートされています。ネットマスクは 0 ～ 127 の整数である必要があります。

列の順序は重要ではありません。最初の 32 のユーザー定義列は、ラベルが自動的に有効になります。32 を超える列がアップロードされている場合は、ページの右側にあるチェックボックスを使用して、最大 32 の列を有効にすることができます。

カスタムラベルのアップロード

次の手順では、**サイト管理者**、**カスタマーサポート**、またはルート**範囲所有者**の各ロールを持つユーザーがラベルをアップロードする方法を説明します。

始める前に

カスタムラベルをアップロードするには、「ラベルファイルのアップロードに関するガイドライン」セクションに従って、CSV ファイルを作成します。

手順

ステップ 1 左側のペインで、**[整理（Organize）]** > **[ユーザー定義ラベルのアップロード（User Defined Label Upload）]** > **[CSVのアップロード（CSV Upload）]** を選択し、[新しいラベルのアップロード（Upload New Labels）] で [ファイルの選択（Select File）] をクリックします。

ステップ 2 左側のペインで、**[整理（Organize）]**>**[ラベル管理（Label Management）]** を選択し、[新しいラベルのアップロード（Upload New Labels）] で [ファイルの選択（Select File）] をクリックします。

ステップ 3 操作（[追加（Add）]、[マージ（Merge）]、または [削除（Delete）]）を選択します。

- [追加（Add）]：ラベルを新規および既存のアドレス/サブネットに追加します。既存のラベルの代わりに新しいラベルを選択して、競合を解決します。たとえば、データベース内の住所のラベルが *{"foo": "1", "bar": "2"}* で、CSV ファイルに *{"z": "1", "bar": "3"}* が含まれている場合、add` は、このアドレスのラベルを *{"foo": "1", "z": "1", "bar": "3"}* に設定します。
- [マージ（Merge）]：ラベルを既存のアドレス/サブネットにマージします。空の値の代わりに空でない値を選択することで、競合を解決します。たとえば、データベース内のアドレスのラベルが *{"foo": "1", "bar": "2", "qux": "", "corge": "4"}* で、CSV ファイルに *{"z": "1", "bar": "", "qux": "3", "corge": "4-updated"}* が含まれている場合、add` は、このアドレスのラベルを *{"foo": "1", "z": "1", "bar": "2", "qux": "3", "corge": "4-updated"}* に設定します。

  （注） “bar” の値は “”（空）にリセットされず、既存の値 “bar”=”2” が保持されます。

- [削除（Delete）]：このオプションにより、アドレス/サブネットのラベルが削除されます。このオプションは、範囲、フィルタ、ポリシー、および適用される動作に大きな影響を与える可能性があります。重要な詳細については、「ラベルの削除」を参照してください。

重要：この削除機能により、カスタムラベルのアップロード中に、指定された IP アドレス/サブネットに関連付けられたすべてのラベルが削除されます。削除の対象は、CSV ファイルに一覧表示されている列に限定されません。そのため、削除操作は慎重に使用する必要があります。

ステップ 4 [アップロード（Upload）] をクリックします。

検索ラベル

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）] または [範囲所有者（scope owner）] のロールを持つユーザーは、特定の IP アドレスまたはサブネットにラベルを割り当てることができます。

手順

ステップ 1 [ユーザーがアップロードしたラベル（User Uploaded Labels）] ページで、[検索と割り当て（Search and Assign）] をクリックします。

ステップ 2 [ラベル管理（Label Management）] ページで、[検索と割り当て（Search and Assign）] をクリックします。

ステップ 3 [IPまたはサブネット（IP or Subnet）] フィールドで、IP アドレスまたはサブネットを入力し、[次へ（Next）] をクリックします。

[ラベルの割り当て（Assign Labels）] ページに、入力した IP アドレスまたはサブネットの既存のラベルが表示されます。

カスタムラベルの手動割り当てまたは編集

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）] または [ルート範囲の所有者（Root Scope Owner）] のロールを持つユーザーは、特定の IP アドレスまたはサブネットにラベルを手動で割り当てることができます。

手順

ステップ 1 [ユーザーがアップロードしたラベル（User Uploaded Labels）] ページで、[検索と割り当て（Search and Assign）] をクリックします。

ステップ 2 [ラベル管理（Label Management）] ページで、[検索と割り当て（Search and Assign）] をクリックします。

ステップ 3 [IPまたはサブネット（IP or Subnet）] フィールドで、IP アドレスまたはサブネットを入力し、[次へ（Next）] をクリックします。

[ラベルの割り当て（ Assign Labels）] ページが表示されます。既存のラベルが表示され、編集できることに注意してください。

ステップ 4 新しいラベルを追加するには、**Labels for <IP address/subnet>** セクションで、ラベル名と値を入力し、[確認（Confirm）] をクリックします。[次へ（Next）] をクリックします。

ステップ 5 変更を確認し、[割り当て（Assign）] をクリックして確定します。

ラベルのダウンロード

サイト管理者、**カスタマーサポート**、またはルート**範囲の所有者**ロールを持つユーザーは、ルート範囲に属する事前に定義されたラベルをダウンロードできます。

手順

ステップ 1 [ユーザーがアップロードしたラベル（User Uploaded Labels）] ページで、[CSVアップロード（CSV Upload）] タブをクリックします。

ステップ 2 [ラベル管理（Label Management）]ページで、[ユーザー定義ラベルのアップロード（User Defined Label Upload）] をクリックします。

ステップ 3 [既存のラベルのダウンロード（Download Existing Labels）] セクションで、[ラベルのダウンロード（Download Labels）] をクリックします。

Secure Workloadで 使用されるラベルが CSV ファイルとしてダウンロードされます。

ラベルの変更



警告 ラベルを変更する必要がある場合は、慎重に行ってください。変更すると、既存のクエリ、フィルタ、範囲、クラスタ、ポリシー、およびそのラベルに基づいて適用された動作のメンバーシップと効果が変更されます。

手順

ステップ 1 [ユーザーがアップロードしたラベル（User Uploaded Labels）] ページで、[検索と割り当て（Search and Assign）] タブをクリックします。

ステップ 2 [ラベル管理（Label Management）] ページで、[検索と割り当て（Search and Assign）] タブをクリックします。

ステップ 3 [IPまたはサブネット（IP or Subnet）] フィールドで、IP アドレスまたはサブネットを入力し、[次へ（Next）] をクリックします。

Secure Workload が入力した IP アドレス/サブネットに対して使用しているラベルが表示されます。

ステップ 4 [アクション（Actions）] 列で、[編集（Edit）] アイコンをクリックして必要なラベルの名前と値を変更します。

ステップ 5 [確認（Confirm）] をクリックし、[次へ（Next）] をクリックします。

ステップ 6 変更を確認し、[割り当て（Assign）] をクリックします。

ラベルの無効化

スキーマを変更する 1 つの方法は、ラベルを無効にすることです。注意して続行してください。

手順

ステップ 1 [ユーザーアップロードラベル（User Uploaded Labels）] ページで、[ラベル（Labels）] タブをクリックします。

ステップ 2 [ラベル管理（Label Management）] ページに移動します。

ステップ 3 該当するラベルの [アクション（Actions）] 列で [無効化（Disable）] を選択し、次に [はい（Yes）] をクリックしてインベントリからラベルを削除することを確認します。

後でラベルを有効にする場合は、[有効化（Enable）] をクリックしてラベルを使用します。

ラベルを削除する

警告 スキーマを変更する1つの方法は、ラベルを無効にして削除することです。注意して続行してください。この操作により、選択したラベルが削除され、依存するすべての**フィルタ**と**範囲**に影響します。これらのラベルが使用されていないことを確認してください。この操作を取り消すことはできません。

手順

ステップ 1 ラベルを無効にします。disable_lables を参照してください。

ステップ 2 [ゴミ箱（TrashCan）] アイコンをクリックし、[はい（Yes）] をクリックしてラベルの削除を確定します。

ラベルの使用状況の表示

IP アドレスやサブネットインベントリは、CSV ファイルを使用してアップロードされたり、ユーザーによって手動で割り当てられたカスタムラベルで更新されます。次に、ラベルは範囲とフィルタの定義に使用され、このフィルタに基づいてアプリケーションポリシーが作成されます。したがって、ラベルを変更すると、Cisco Secure Workload の範囲、フィルタ、およびポリシーに直接影響するため、ラベルの使用状況を把握することが重要です。

ラベルの使用状況を表示するには、次の手順を実行します。

手順

ステップ 1 [ユーザーアップロードラベル（User Uploaded Labels）] ページで、[ラベル（Labels）] タブを選択します。ラベルキー、使用中のラベルの上位 5 つの値、インベントリ、範囲、フィルタ、およびカスタムラベルを使用するクラスタが表示されます。

ステップ 2 [ラベル管理（Label Management）] ページでは、ラベルキー、使用中のラベルの上位 5 つの値、インベントリ、範囲、フィルタ、およびカスタムラベルを使用するクラスタが表示されます。

ステップ 3 [使用状況（Usages）] 列で、インベントリ、範囲、またはフィルタに対するカウント値をクリックします。たとえば、[ロケーション（Location）] ラベルを使用している範囲を表示するには、範囲クエリ数をクリックします。

図 *173* : 選択したラベルの範囲を表示

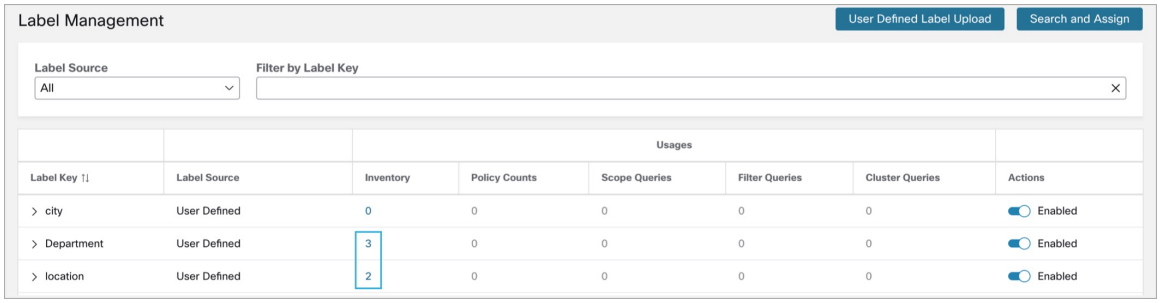
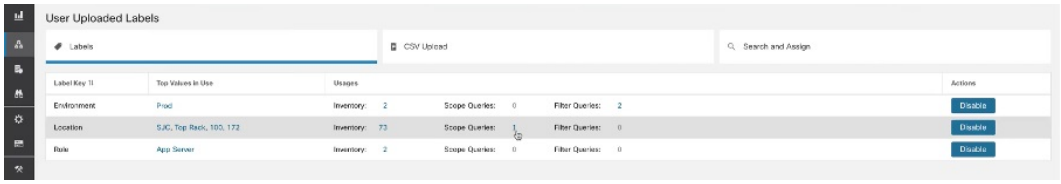


図 *174* : 選択したラベルの範囲を表示



[範囲とインベントリ（Scopes and Inventory）] ページが表示され、クエリによって、選択したラベルで範囲が自動的にフィルタ処理されます。

（注） CSVファイルを使用してアップロードされたラベル、またはIPアドレスやサブネットに手動で割り当てられたラベルの使用状況のみを表示できます。

ラベルのメンテナンスプロセスの作成

ネットワークとインベントリは変更されるため、ラベルを更新して変更を反映する計画を立てる必要があります。

たとえば、あるワークロードが廃止され、その IP アドレスが別の目的のワークロードに再度割り当てられた場合、そのワークロードに関連付けられているラベルを更新する必要があります。これは、手動でアップロードされるラベルと、構成管理データベース（CMDB）などの他のシステムで維持され、取り込まれるラベルの両方に当てはまります。

ラベルが定期的に継続して更新されるプロセスを作成し、そのプロセスをネットワーク メンテナンス ルーチンに追加します。

範囲とインベントリ

範囲とインベントリの概要

このセクションでは、範囲階層とそれに含まれるすべてのインベントリについて明らかにします。範囲は、階層構造を使用してすべてのインベントリを分類します。インベントリ（407ページ）を参照してください。左側は範囲ディレクトリのユーザーインターフェイスです。ここで、範囲階層を下位方向に移動できます。各範囲は範囲カードに表示されます。範囲の名前が表示されます。該当する場合は、子範囲の数、インベントリ数、未分類のインベントリも表示されます。範囲カードをクリックすると、右側のペインが更新され、その範囲に関する詳細と、そのすべてのインベントリのフィルタ可能なリストが表示されます。

範囲設計の原則

1. 動的クエリ一致に従って、インベントリが範囲ツリーに対して照合されます。
   - クエリは、IP/サブネットまたはラベル（推奨）に対して照合されます。
   - ツリーは、各層におけるクエリの結合によって形成されます。
2. 必要に応じて、範囲構造はロケーションに固有にすることができます。
   - クラウドの結合に対するデータセンター、クラウド固有に対する地理的ロケーション
3. 範囲ツリーの各レイヤーは、次のアンカーポイントを意味します。
   - ポリシー制御
   - ロール ベース アクセス制御（RBAC）（Role Based Access Control）

4. すべての子範囲は、その親範囲のサブセットである必要があります
   - 兄弟の範囲が重複していないことを確認してください（「範囲の重複」を参照）。

（注） すべての組織の構造は異なるため、業界に応じて異なるアプローチが必要です。範囲の階層を設計する際に、ロケーション、環境、またはアプリケーションといった注目点を1つ選択します。

（注） IP アドレスまたはサブネットを使用して、Kubernetes インベントリに関連する範囲を定義しないでください。これらのワークロードの範囲とポリシーの定義には、ラベルを使用する必要があります。Pod サービスを識別するには、IP アドレスだけでは不十分です。範囲定義に IP アドレスを使用すると、信頼性の低い結果が生成されます。

主な機能

範囲とインベントリの両方のフィルタリング機能により、範囲ツリーをすばやく下方向に移動したり、範囲階層をフィルタリングして選択した範囲のインベントリ項目をフィルタリングしたりすることができます。

インベントリ数は範囲カードに表示され、範囲内のワークロードの量をすばやく確認できます。

スコープ

範囲は、Cisco Secure Workload の構成とポリシーの基本的な要素です。範囲とは、階層に配置された一群のワークロードです。ワークロードは、位置、ロール、および環境内の機能に関するモデルを構築する属性として機能するようにラベル付けされます。範囲は、時間の経過とともに変化し得る IP に関連付けられた ID や属性などの動的メカニズムをサポートする構造を提供します。

範囲はデータセンターアプリケーションをグループ化するために使用され、ロールを使用してそれらの管理をきめ細かく制御できるようにします。たとえば、範囲は ポリシー （513 ページ）、フロー、フィルタへのアクセスを定義するために製品全体で使用されます。

範囲は、[VRF] に対応するルートを持つ一連のツリーとして階層的に定義されます。その結果、それぞれの範囲ツリー階層は、別の範囲ツリーと重複しない分離されたデータを表示します。「範囲の重複」を参照してください。

範囲の定義

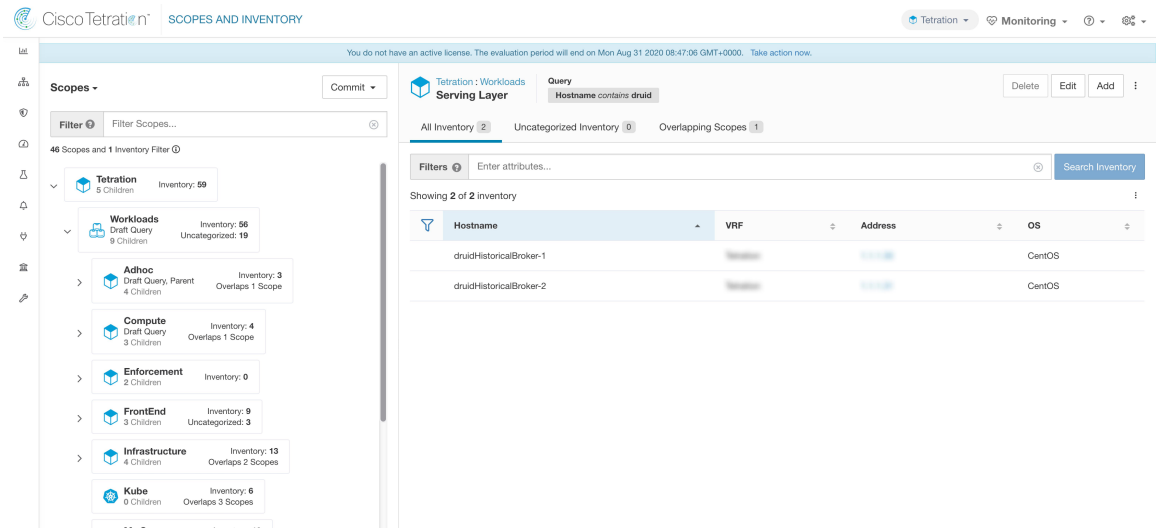
個々の範囲は、次の属性で定義されます。

属性	説明
[親範囲（Parent Scope）]	新しい範囲の親は、ツリー階層構造を定義します。
名前	範囲を識別する名前。
タイプ	タイプは、さまざまなカテゴリのインベントリを指定するために使用されます。該当がない場合、または範囲に混合タイプが含まれている場合は、空白のままにすることができます。
Query	個々の範囲を定義するクエリ。

（注） 範囲は、組織のアプリケーション所有権階層を模倣する階層で定義する必要があります。

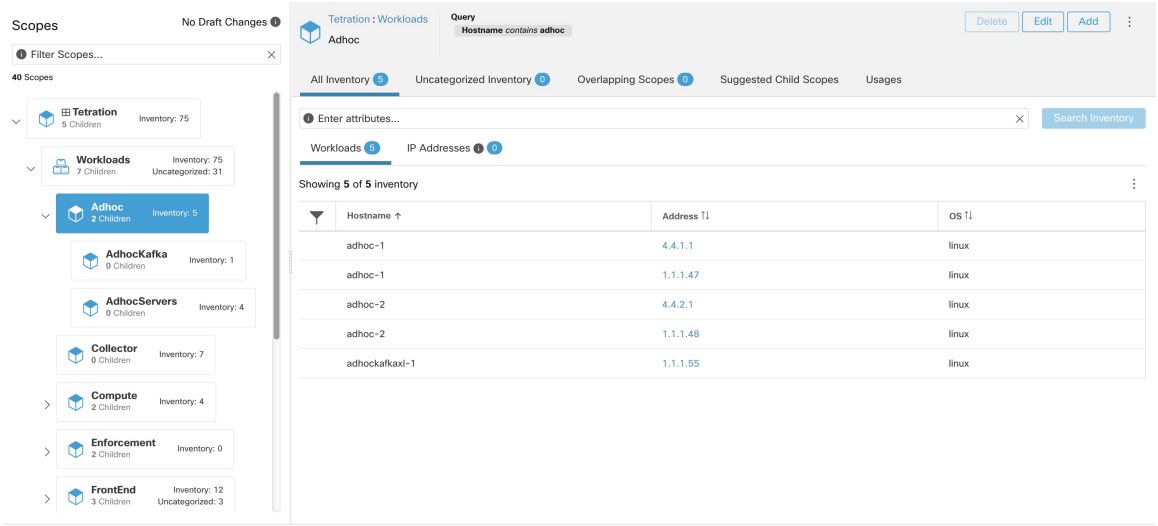
（注） クエリは、IP やサブネットまたはその他のインベントリ属性と一致させる場合があります。

図 *175*：範囲階層を横断する例



範囲ディレクトリには、範囲階層と各範囲の詳細（インベントリ数、子範囲の数、ワークスペースなど）が表示されます。範囲をクリックするとその範囲が選択されます。右側の詳細ペインが、範囲およびその範囲のインベントリに関する詳細情報で更新されます。

図 *176* : インベントリ カウント



範囲フィルタ

ユーザーは、範囲フィルタを使用して、範囲の重複やクエリなど、さまざまな範囲の詳細をすばやく特定できます。フィルタ機能は、クエリの変更、親の変更などを特定するのにも役立ちます。

フィールド	説明
名前	範囲またはインベントリフィルタの名前でフィルタ処理します。
説明	範囲の説明に表示されるテキストでフィルタ処理します。
Query	クエリで使用されるフィールドまたは値でフィルタ処理します。
[クエリの変更（Query Change）]	コミットされていないクエリがある範囲でフィルタ処理します。
Parent Change	ドラフトで移動されたがコミットされていない範囲でフィルタ処理します。
[インベントリフィルタあり（Is Inventory Filter）]	所有権の範囲に制限されているインベントリフィルタを表示します。
[ワークスペースあり（Has Workspace）]	プライマリワークスペースがある範囲でフィルタ処理します。
[適用ワークスペースあり（Has Enforced Workspace）]	適用されているプライマリワークスペースがある範囲でフィルタ処理します。

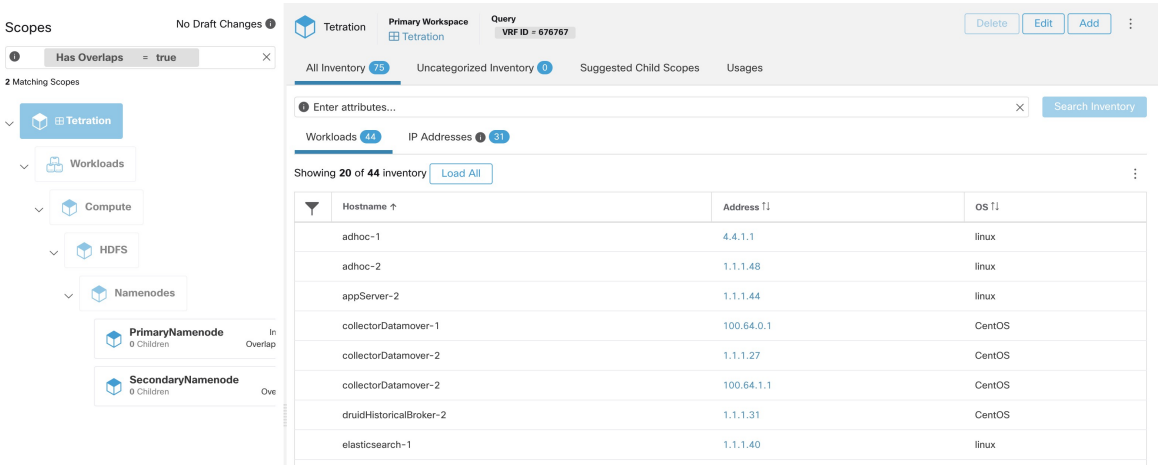
フィールド	説明
[重複あり（Has Overlaps）]	兄弟範囲と共通のインベントリがある範囲でフィルタ処理します。
[無効なクエリあり（Has Invalid Query）]	無効または不明なラベルを使用するクエリを含む範囲でフィルタ処理します。

例：

[重複あり（Has Overlaps）]

範囲の重複の例

図 *177 : Has Overlaps*

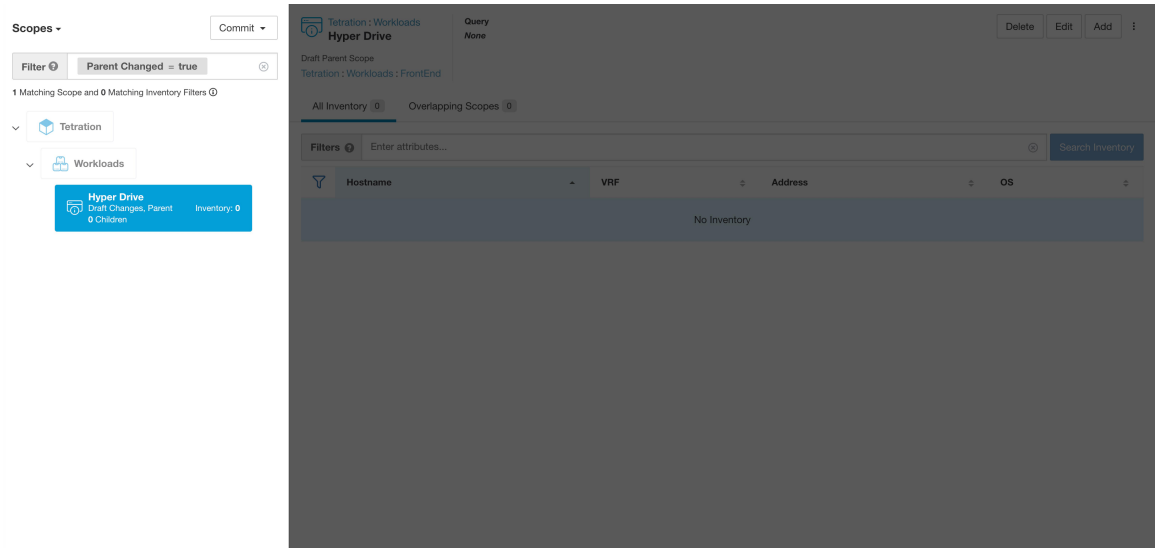


詳細については、「範囲の重複」の項を参照してください。

Parent Change

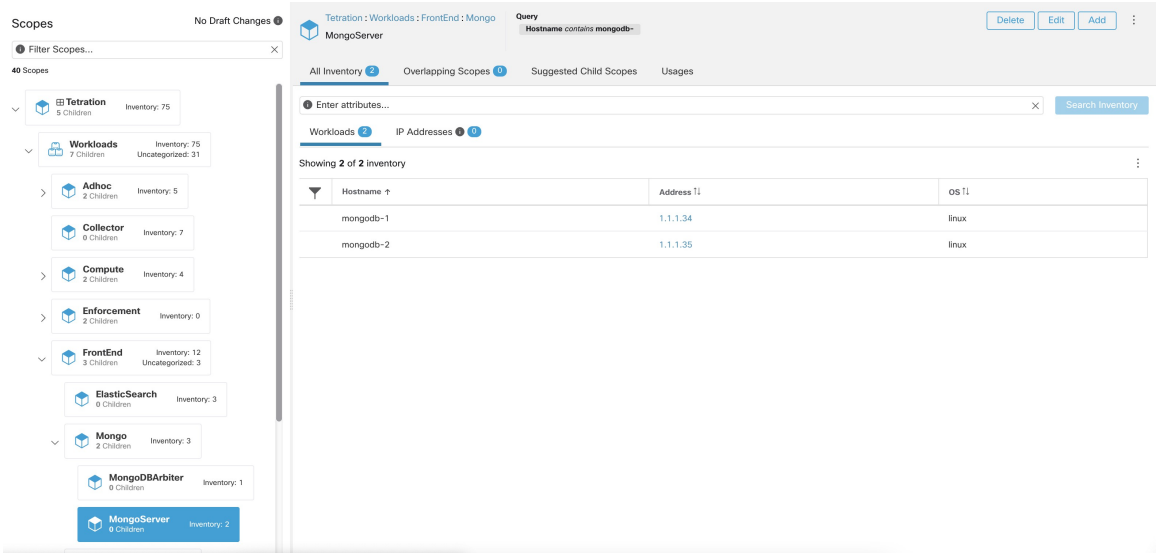
ドラフトで移動されたがまだコミットされていない範囲。

図 *178 : Parent Change*



フル範囲のクエリ

図 *179 :* 範囲階層の例



範囲の定義は階層型になっており、範囲のフルクエリは、すべての親と共に範囲の「論理積」として定義されます。上記の例では、アセットは `Workloads:FrontEnd:Mongo` に割り当てられています。

範囲の照合は次のようになります。

```
vrf_id = 676767 and (ip in 1.1.1.0/24) and (Hostname contains mongo).
```

vrf_id = 676767 はルート範囲クエリに基づき、1.1.1.0/24 の IP は親範囲クエリに基づきます。

（注） 同じレベルでクエリが重複しないようにすることを推奨します。これにより、順序付けの重要性がなくなり、混乱が軽減されます（「範囲の重複」を参照）。

範囲へのアクセスの提供

ユーザーには、範囲上での読み取り、書き込み、実行、適用、および所有者の権限が与えられます。概要を以下に示します。完全な情報については「ロール」を参照してください。

ユーザーには「サブツリー」（与えられた範囲およびそのすべての子範囲など）へのアクセス権が与えられます。上記の例を使用すると、`Workloads:FrontEnd`範囲への読み取りアクセス権があるユーザーは、継承により、以下を含む`Workloads:FrontEnd`の下のすべての範囲への読み取りアクセス権があります。

- `Workloads:FrontEnd:Mongo`
- `Workloads:FrontEnd:ElasticSearch`
- `Workloads:FrontEnd:Redis`
- `etc. . .`

複数の範囲にアクセス可能なロールを定義できます。たとえば、「Mongo Admin」ロールには、範囲への所有者アクセス権がある場合があります。

- `Workloads:FrontEnd:Mongo:MongoServer`
- `Workloads:FrontEnd:Mongo:MongoDBArbiter`

ロールと機能により、ユーザーは範囲階層への「水平な」アクセス権を持つことができます。

範囲の機能も継承されます。たとえば、範囲に書き込み機能があると、その情報の読み取りも可能です。

範囲の表示

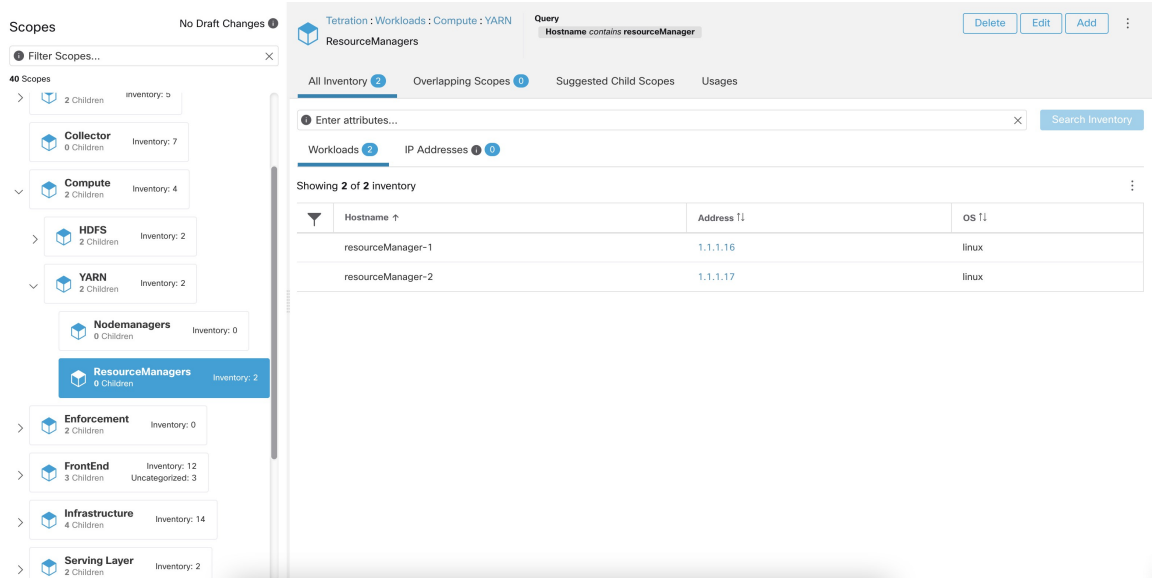
すべてのユーザーは、自分にアクセス権がある範囲ツリーを表示できます。ルート範囲の所有者権限を持つユーザーは、そのツリーの範囲を作成、編集、および削除できます。このビューにアクセスするには、以下を行います。

左側のナビゲーションバーで、**[整理（Organize）]** > **[範囲とインベントリ（Scopes and Inventory）]**をクリックします。

アクセス権のあるすべての範囲について、範囲階層全体（ルートまで）を横断できます。この完全な横断により、ユーザーが任意の範囲に対するポリシーを作成する際のコンテキストが提供されます。このページでは、いくつかのアクションを実行できます。

- 範囲階層の V 字の部分をクリックすると、その範囲の子が表示されます。
- 範囲カードをクリックすると、右側のペインが更新され、その範囲に関する詳細と、そのすべてのインベントリのフィルタ可能なリストが表示されます。

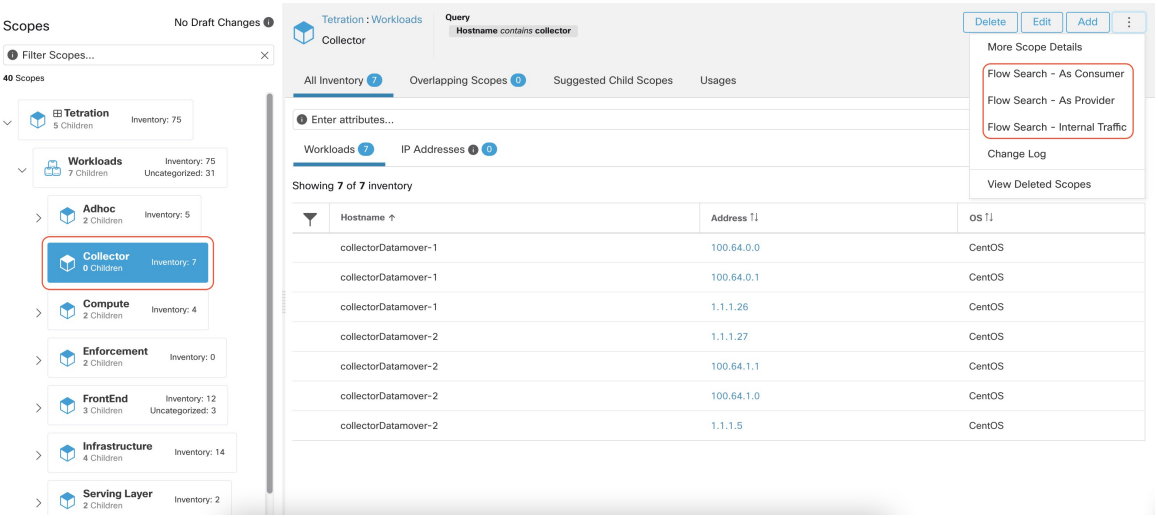
図 *180* : 非管理者ビューの例



範囲を参照するフローの検索

範囲ページには、ショートカットがいくつか用意されており、フローの一方または両方のエンドポイントが指定された範囲内にあるフローを検索する必要があるシナリオで役立ちます。

図 *181* : 範囲でのフローの検索



上の図に示すように、範囲ツリー（左側のパネル）で目的の範囲を選択した後、ユーザーは次の 3 つのオプションから選択できます。

1. [フロー検索（コンシューマとして）（Flow Search - As Consumer）] では、フロー検索ページへのショートカットが提供され、フローのコンシューマ範囲として選択された範囲でのフローの検索に役立ちます。つまり、フローのコンシューマエンドポイントまたは送信元エンドポイントは、選択した範囲に属します。

2. [フロー検索 （プロバイダーとして）（Flow Search - As Provider）] では、フロー検索ページへのショートカットが提供され、フローのプロバイダー範囲として選択された範囲でのフローの検索に役立ちます。つまり、フローのプロバイダーエンドポイントまたは宛先エンドポイントは、選択した範囲に属します。
3. [フロー検索 （内部トラフィック）（Flow Search - Internal Traffic）] では、フロー検索ページへのショートカットが提供され、選択した範囲に完全に制限されているフローの検索に役立ちます。つまり、フローの両方のエンドポイント（コンシューマとプロバイダー）は、選択した範囲に属します。

新しい範囲の作成

子範囲は、[範囲（Scopes）] 管理ページで作成します。このアクションには、ルート範囲の SCOPE_OWNER 権限が必要です。**サイト管理者**は、すべての範囲の所有者です。

子範囲を作成すると、親範囲のアプリケーションインベントリ メンバーシップ（メンバーワークロード）に影響します。その結果、親範囲には「ドラフト変更」のマークが付けられます。変更をコミットして、依存関係の構造を更新する必要があります。「変更の確定」を参照してください。

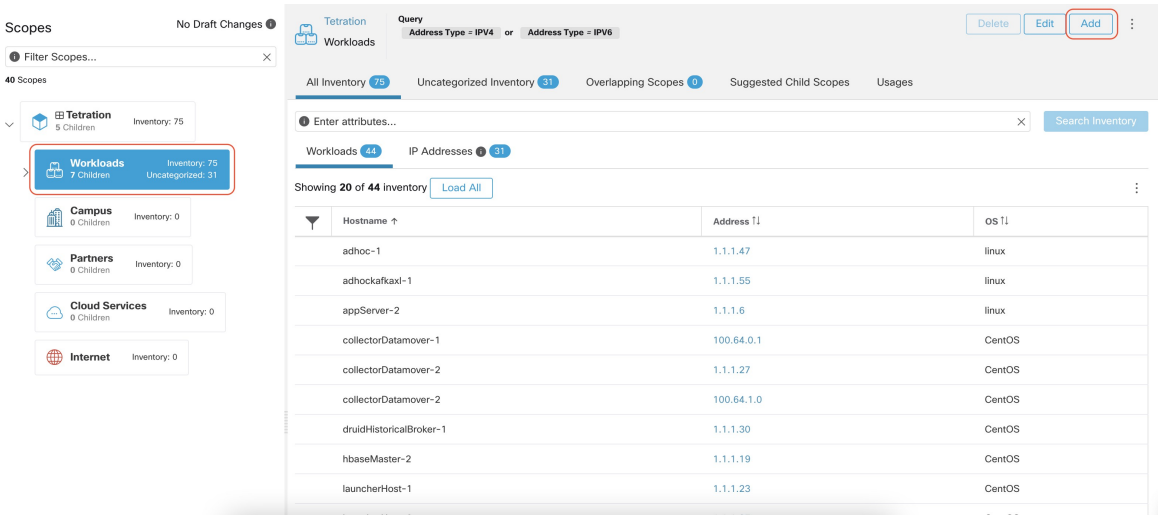
手順

ステップ 1 左側のナビゲーションバーで、**[整理（Organize）] > [範囲とインベントリ（Scopes and Inventory）]**をクリックします。このページには、システムで作成済みのテナントと VRF に対応するルート範囲が表示されます。

ステップ 2 範囲ディレクトリで子範囲を選択します。必要に応じて、最初に範囲をフィルタリングできます。

ステップ 3 [追加（Add）] ボタンをクリックします。

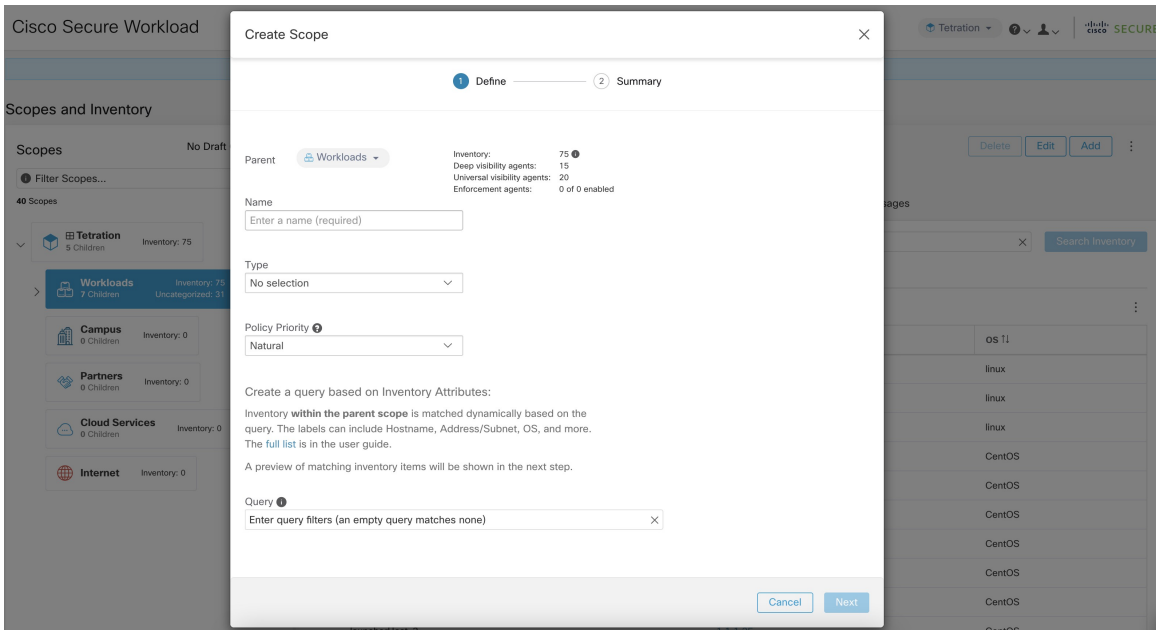
図 *182 : [*範囲の追加（*Scope Add***）*]*ボタン**



ステップ 4 以下のフィールドに適切な値を入力します。

フィールド	説明
[親（Parent）]	新しい範囲の親。
名前	範囲を識別する名前。親範囲の下で一意である必要があります。
[タイプ（Type）]	新しい範囲のカテゴリを選択します。
[クエリ（Query）]	アセットに一致するクエリまたはフィルタ。

図 *183 :* 範囲作成モーダル

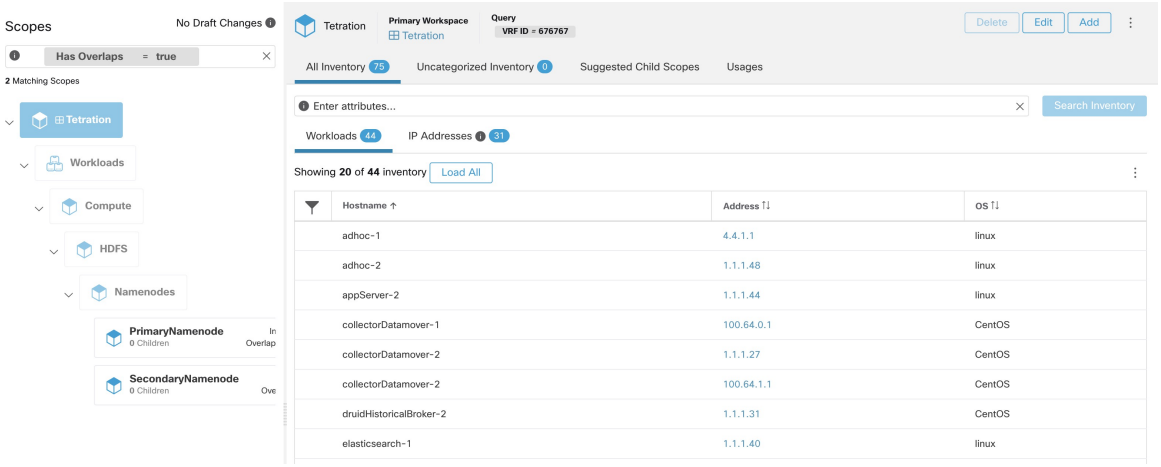


範囲の重複

範囲を追加するときは、範囲の重複を避けることをお勧めします。範囲が重複すると、重複する範囲に対して生成されたポリシーが、エンドユーザーの混乱を招く可能性があります。この機能は、範囲メンバーシップが重複している場合、つまり、同じインベントリが範囲ツリー（兄弟範囲）の同じ深さの複数の範囲に属している場合に、積極的にユーザーに通知します。その目的は、範囲ツリーの異なる部分に同じワークロードが存在することを避けることです。

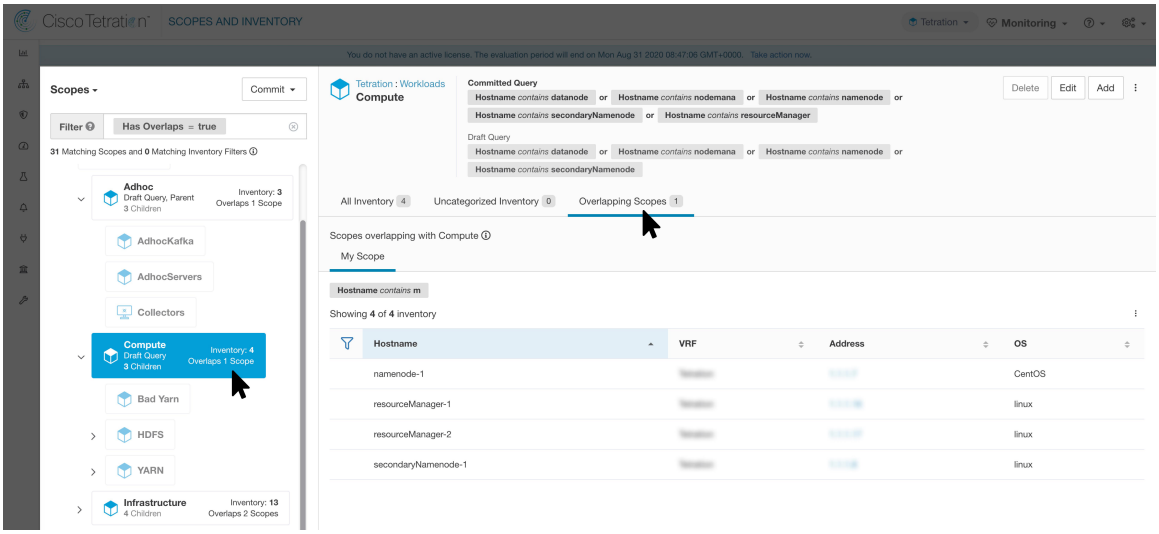
複数の範囲に属するインベントリ項目を表示するには、範囲フィルタを使用し、**Has Overlaps = true** ファセットを入力します。

図 *184* : 範囲フィルタでの重複ファセット



重複する範囲および対応する重複する IP アドレスのリストは、範囲ツリーを下に移動し、[重複する範囲（Overlapping Scopes）] タブを選択することで表示できます。

図 *185* : 範囲と *IP* の重複



スコープの編集

ルート範囲で SCOPE_OWNER 権限を持つユーザーのみが範囲を削除できます。サイト管理者は、すべての範囲の所有者です。

範囲名の編集

範囲名の編集は瞬時に完了しますが、更新が必要な子範囲の数によっては数分かかる場合があります。

（注） 範囲名を変更すると、範囲名によるフロー検索が影響を受けます。

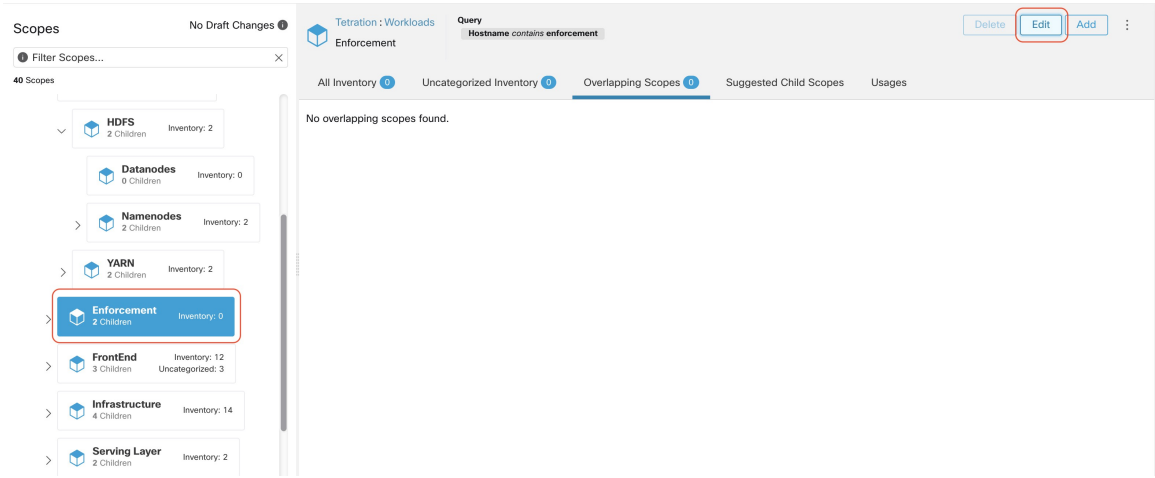
範囲クエリの編集

範囲のクエリが変更されると、直接の親と子の範囲が影響を受けます。これらの範囲は、確定されていない変更がツリーに加えられたことを示す「ドラフト変更」があるとマークされています。すべてのクエリの更新が完了したら、ユーザーは範囲ディレクトリの上にある [変更の確定（Commit Changes）] ボタンをクリックして、変更を永続化する必要があります。これにより、バックグラウンドタスクがトリガーされ、ワークスペース内のすべての範囲クエリと「動的クラスタクエリ」が更新されます。

警告 範囲クエリを更新すると、範囲のインベントリメンバーシップ（範囲のメンバーであるワークロード）に影響を与える可能性があります。変更は、変更の確定プロセス中に有効になります。リスクを軽減するために、[範囲/フィルタ変更の影響を確認する（Review Scope/Filter Change Impact）] ウィンドウから、メンバーシップの変更を比較して、影響を詳細に分析できます。範囲/フィルタ変更の影響を確認 （455 ページ）

新しいホスト ファイアウォール ルールが挿入され、関連するホスト上で既存のルールがすべて削除されます。

図 *186 :* 範囲の編集



範囲を編集するには、次の手順を実行します。

手順

ステップ 1 編集するそれぞれの範囲の**編集ボタン**をクリックします。

ステップ 2 選択した範囲の名前またはクエリを編集します。

ステップ 3 [クエリ変更の影響の確認（Review query change impact）] のリンク先に移動して、古いドラフトクエリと新しいドラフトクエリの間の変更を比較します。

ステップ 4 [保存（Save）] をクリックします。名前はすぐに更新されます。

ステップ 5 すべての範囲のクエリを更新するには、[変更の確定（Commit Changes）] ボタンをクリックします。

ステップ 6 範囲の変更を実行した結果を示す確認ポップアップウィンドウが表示されます。更新は、バックグラウンドタスクで非同期に処理されます。

ステップ 7 [保存（Save）] をクリックします。変更の数によっては、保存に 1 分以上かかる場合があります。

図 *187* : クエリ変更の影響の確認

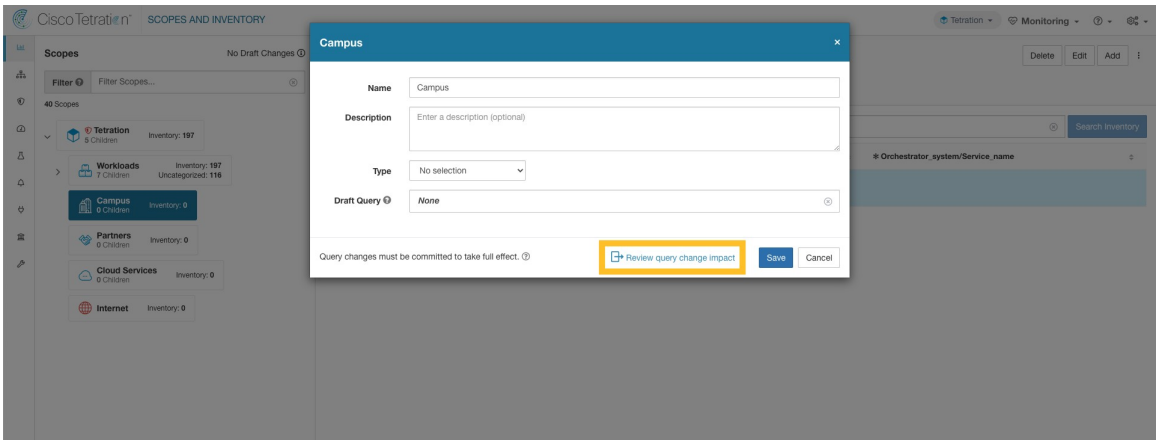
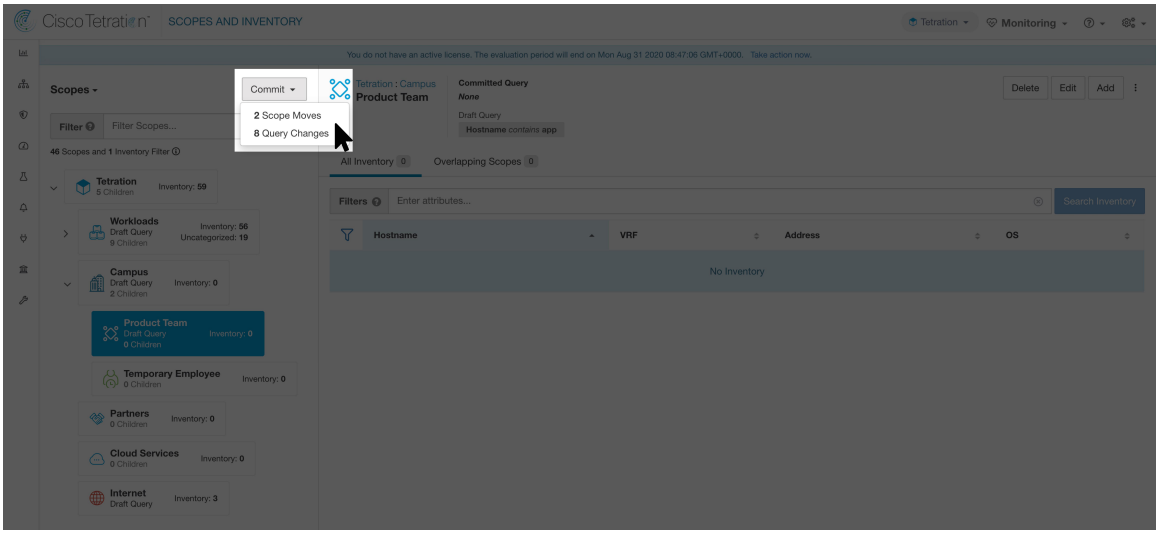


図 *188* : 変更の確定



範囲の親の編集

範囲の親が更新されると、範囲クエリが変更されます。この変更は、親スコープと子スコープの両方のメンバーシップに影響します。範囲クエリの編集と同様に、これらの変更は最初「ドラフト変更」として保存され、確定されない限り有効になりません。ユーザーは、[範囲の編集（Edit Scope）] モーダルで [クエリ変更の影響を確認（Review query change impact）] をクリックして、確定する前にこの変更の影響を検証できます。検証が完了したら、[確定（Commit）] をクリックし、[範囲の移動（Scope Moves）] と [クエリの変更（Query Changes）] を承認することで、変更を確定できます。

範囲の親を編集するには、次の手順に従います。

手順

ステップ 1 編集するそれぞれの範囲の [編集（Edit）] ボタンをクリックします。

ステップ 2 選択した範囲の親を編集します。

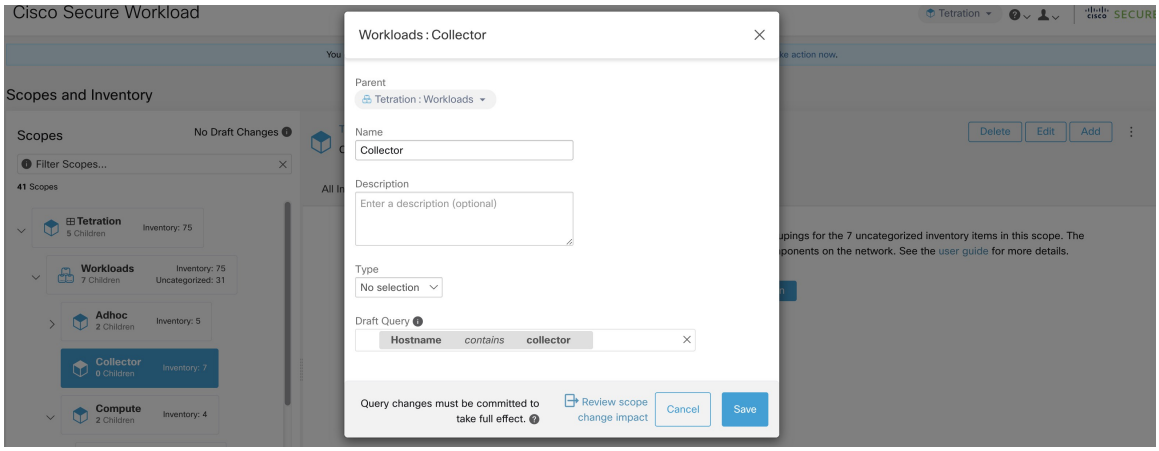
ステップ 3 [クエリ変更の影響の確認（Review query change impact）] リンクをクリックして、古いドラフトクエリと新しいドラフトクエリの間の変更点を比較します。

ステップ 4 [保存（Save）] をクリックします。

ステップ 5 [確定（Commit）] をクリックし、[範囲の移動（Scope Moves）] と [クエリの変更（Query Changes）] を承認します。更新は、バックグラウンドタスクで非同期的に処理されます。

ステップ 6 この変更の影響を受けるワークロードの数によっては、処理に 1 分以上かかる場合があります。

図 *189 :* 親範囲をデフォルト範囲から *Default:ProdHosts* に変更する



スコープの削除

ルート範囲で SCOPE_OWNER 機能を持つユーザーのみが範囲を削除できます。サイト管理者は、すべての範囲の所有者です。

範囲を削除すると、親範囲（親範囲のメンバーであるワークロード）のアプリケーションインベントリ メンバーシップに影響があります。. その結果、親範囲は「ドラフト変更」を持つものとしてマークされます。変更をコミットして、依存関係の構造を更新する必要があります。「変更の確定」を参照してください。

依存オブジェクトのある範囲は削除できません。次の場合はエラーが返されます。

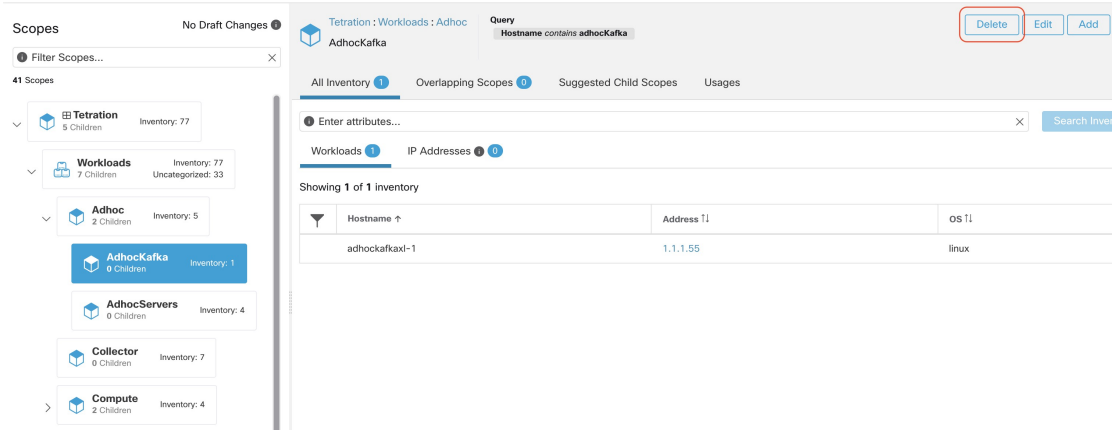
- 範囲にワークスペースが定義されている。
- 範囲に割り当てられたインベントリフィルタがある。
- 範囲を使用してコンシューマまたはプロバイダーを定義するポリシーが存在する。
- 範囲で Agent Config Intent が定義されている。
- 範囲で Interface Config Intent が定義されている。
- 範囲で Forensics Config Intent が定義されている。

範囲の依存関係をさらにドリルダウンするには、「範囲/フィルタ変更の影響を確認」ウィンドウから、[依存関係（Dependencies）] タブにアクセスします。

範囲を削除する前に、これらのオブジェクトを削除する必要があります。

1. 左側のナビゲーションバーで、**[整理（Organize）]>[範囲とインベントリ（Scopes and Inventory）]** をクリックします。
2. [範囲（scope）] を選択して再度クリックすると、子範囲が表示されます。削除する子範囲を選択します。
3. [編集（edit）] ボタンと [追加（add）] ボタンの横にある [削除（Delete）] ボタンをクリックします。

図 *190* : 範囲の削除



（注） 子のない範囲のみ削除できます

（注） [テナント（Tenants）] ページから VRF を削除して、ルート範囲を削除する必要があります。

範囲ツリーのリセット

上記の構成のいずれかが存在する場合は、範囲ツリーをリセットする前にそれらを削除する必要があります。これを行うまで、[リセット（Reset）] ボタンは使用できません。

範囲ツリーをリセットするには：

始める前に

範囲ツリー全体を削除して、最初からやり直すことができます。

範囲ツリーをリセットすると、すべての範囲、ラベル、ワークスペース、およびコレクションルールが削除されます。取り込まれたデータは削除されません。

ルート範囲で SCOPE_OWNER 機能を持つユーザーのみが、範囲ツリーをリセットできます。

ただし、ツリー内のいずれかの範囲に対して次のいずれかが定義されている場合、範囲ツリーをリセットすることはできません。

- ワークスペース（ウィザードを使用して範囲ツリーを作成した場合に作成された単一のワークスペースを除く）
- インベントリ フィルタ
- ポリシー
- エージェント構成インテント
- インターフェイス設定インテント
- フォレンジック構成インテント

手順

ステップ 1 左側のナビゲーションメニューから、**[整理（Organize）]** > **[範囲とインベントリ（Scopes and Inventory）]** を選択します。

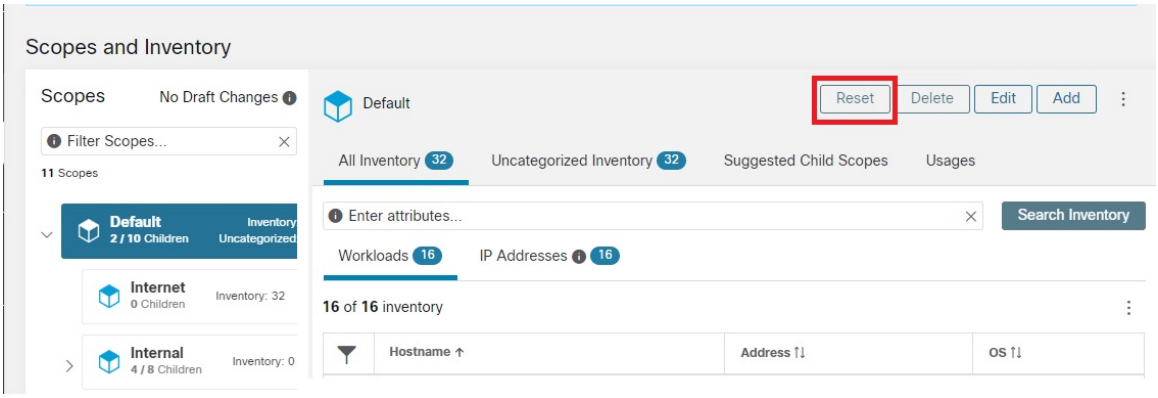
ステップ 2 ツリーの上部にある範囲をクリックします。

ステップ 3 [リセット（Reset）] をクリックします。

ステップ 4 選択を確認します。

ステップ 5 必要に応じて、ブラウザページを更新して続行します。

図 *191* : 範囲ツリーのリセット



変更の確定

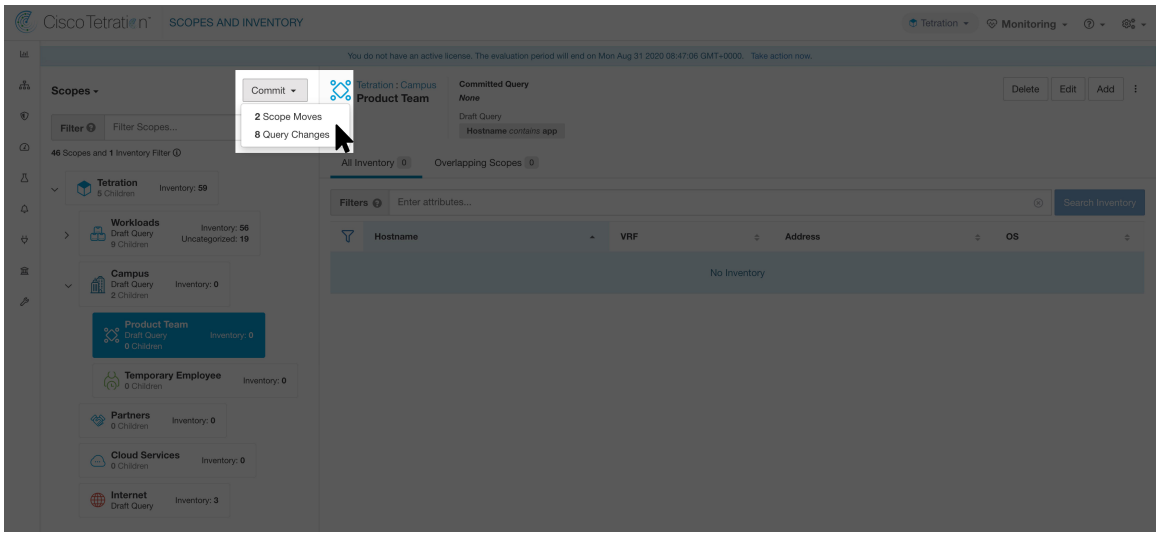
ある範囲のアプリケーション インベントリ クエリ定義は、その範囲のクエリとその直接の子のクエリによって定義されます。これが発生すると、範囲は「ドラフト変更」があるとマークされ、範囲のクエリ、ワークスペース、およびクラスタは、[変更の確定（Commit Changes）] バックグラウンドタスクが実行されるまで変更されません。範囲がドラフトの場合、影響を受ける範囲アイコンごとに三角の注意マークが表示され、[変更の確定（Commit Changes）] ボタンが [範囲（Scopes）] ページに表示されます（右上）。このボタンをクリックして [変更の確定（Commit Changes）] バックグラウンドタスクを実行する必要があります。

範囲をドラフトとしてマークできるイベントは次の通りです。

- クエリの更新
- 親のクエリが更新された
- 直接の子が追加された
- 直接の子が削除された
- 直接の子のクエリが更新された

範囲の名前を変更しても、範囲のドラフト状態は変更されません。

図 *192 :* 変更の確定



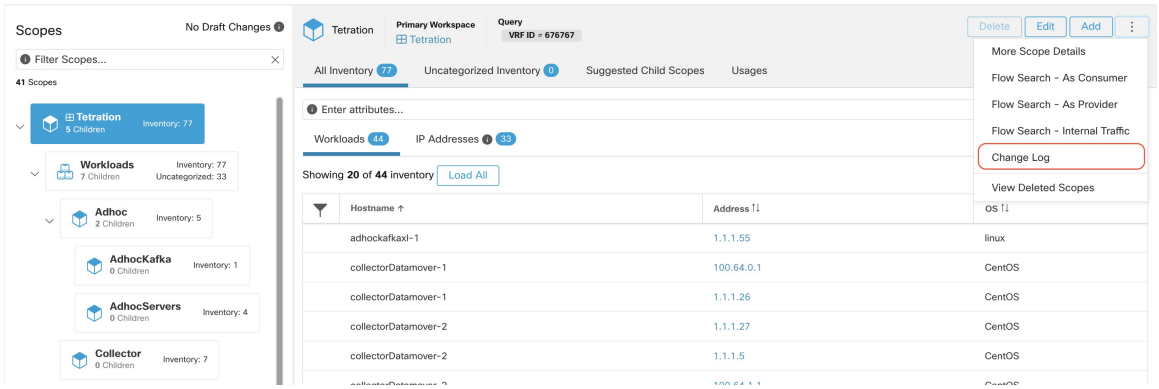
（注） [変更の確定（Commit Changes）] タスクは非同期です。通常は数秒かかりますが、大きな範囲ツリーでは数分かかることがあります。

（注） 範囲の更新タスクは、ルート範囲がドラフトでなくなると完了します。ページを更新して最新の状態を表示します。

ログの変更

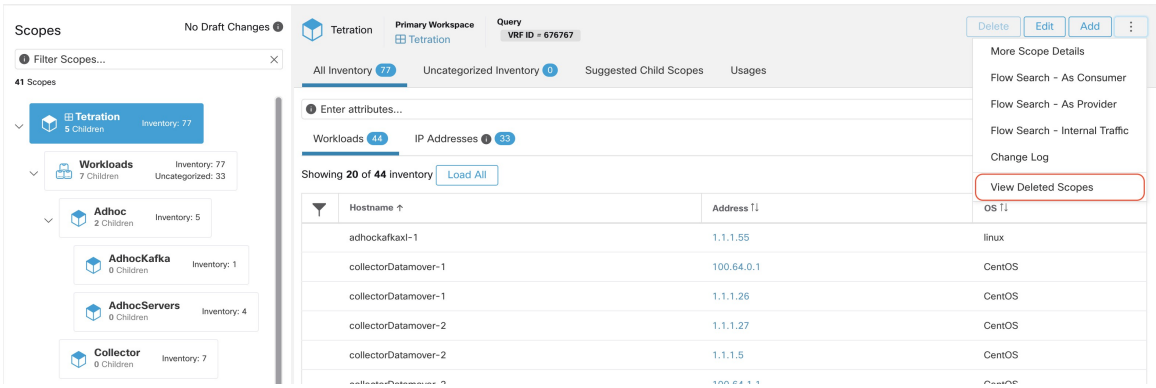
サイト管理者およびルート範囲で SCOPE_OWNER 機能を持つユーザーは、右上のオーバーフローメニューの変更ログをクリックして、各範囲の変更ログを表示できます。

図 *193 :* ログの変更



そのようなユーザーは、右上隅のオーバーフローメニューにある [削除された範囲の表示（View Deleted Scopes）] リンクをクリックして、削除された範囲のリストを表示することもできます。

図 *194* : 削除された範囲の表示（***View Deleted Scopes***）



新しいテナントの作成

ルートレベルの範囲は、テナントまたは [範囲（Scopes）] 管理ページから作成された VRF にマッピングされます。このアクションは、サイト管理者とカスタマーサポートのユーザーのみが利用できます。

手順

ステップ 1 左側のナビゲーションバーで、**[プラットフォーム（Platform）] > [テナント（Tenants）]**をクリックします。

ステップ 2 [新しいテナントの作成（Create New Tenant）] ボタンをクリックします。

ステップ 3 以下のフィールドに適切な値を入力します。

フィールド	説明
名前	範囲を識別する名前。親範囲の下で一意である必要があります。
説明	任意の説明。

ステップ 4 [作成（Create）] ボタンをクリックします。

図 *195 :* テナントの作成（*Create Tenant*）



インベントリ

インベントリで作業するには、左側のナビゲーションバーで**[整理（Organize）]**>**[範囲とインベントリ（Scopes and Inventory）]** をクリックします。

[インベントリ検索（Inventory Search）]

ネットワーク上で検出されたすべてのインベントリが検索可能です。インベントリを検索するには、[インベントリの検索（Search Inventory）] ボタンを使用します。各インベントリ項目は、IP および VRF によって一意に識別でき、検索の実行に使用できます。サービスインベントリ項目は、自身の IP アドレスを使用して検索できません。サービスインベントリを検索するには、user_orchestrator_system/service_name など、サービスに関連付けられているユーザーラベルのいずれかを使用してください。ホストが見つかったら、[ホストプロファイル（host profie）] ページで、ホストに関する詳細情報を表示できます。

インベントリ構成要素

1. ルート範囲
   - 指定されたテナント下の範囲階層ルート
   - L3 アドレスドメインの論理的な分離を提供
2. スコープ
   - 動的クエリで定義されたインベントリコンテナ
   - 階層型ポリシーモデルの基礎
   - ポリシー、RBAC、およびフィルタ設定のアンカーポイント
3. フィルタ
   - 動的なインベントリクエリに基づく柔軟な構築
   - インテント定義、提供サービス、ポリシー定義のアンカーポイント

（注） パートナーからのすべての IP アドレスと、使用中の環境内で通信しているすべてが含まれます。環境にエージェントが存在するかどうかにかかわらず、ラベルを使用して内容を定義する必要があります。

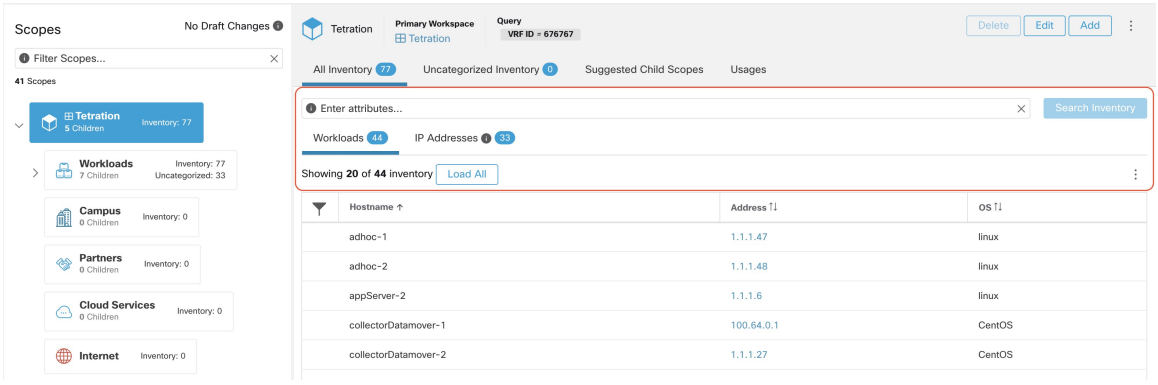
ラベルプランニングの考慮事項

1. データの送信元
   - ネットワークは、IPAM か。ルーティングテーブルか。スプレッドシートか。
   - ホストは、CMDB か。ハイパーバイザか。クラウドか。アプリケーションオーナーか。
2. データの精度
3. データがどの程度動的で、どのように更新されるか
   - 手動アップロードか。
   - API の統合か。
4. 基本から始めて進化させる
   - ネットワークラベルを使用して高度な範囲構造を構築する
   - ホストラベルを使用して、アプリケーションレベルでより詳細な範囲構造を構築する

インベントリの検索

インベントリを検索すると、特定のインベントリ項目に関する情報を表示できます。

図 *196 : [インベントリ検索（Inventory Search）]*



手順

ステップ 1 トップレベルのメニューから、**[整理（Organize）] > [範囲とインベントリ（Scopes and Inventory）]** を選択します。

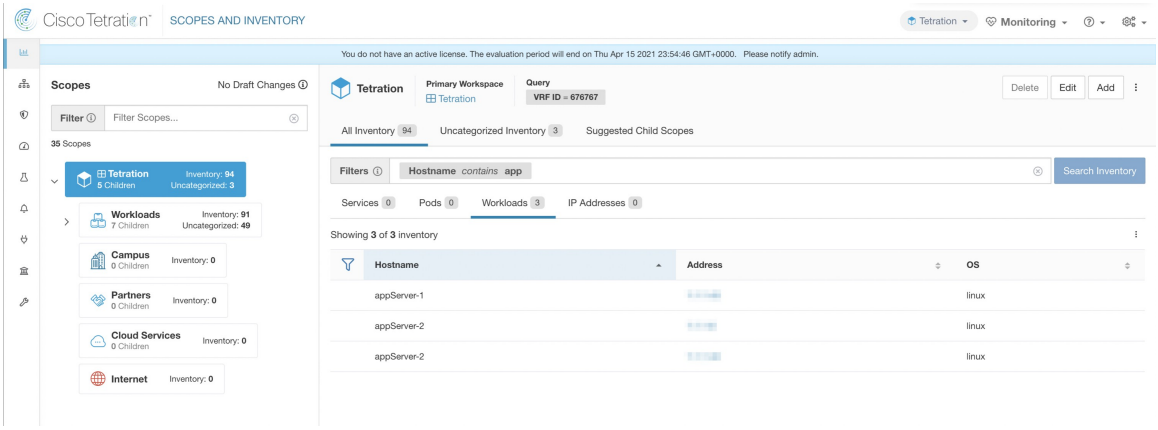
ステップ 2 [フィルタ（Filters）] フィールドに、探しているインベントリ項目の 属性を入力します。次のような属性があります。

属性	説明
ホスト名（Hostname）	ホスト名の全体または一部を入力します。
[VRF名（VRF Name）]	VRF 名を入力します。
VRF ID	VRF ID（数値）を入力します。
アドレス（Address）	有効な IP アドレス（IPv4 または IPv6）を入力します。
アドレス タイプ	IPv4 または IPv6 を入力します。
OS	OS 名（CentOS など）を入力します。
[OS Version]	OS バージョンを入力します（例：6.5）。
Interface Name	インターフェイス名を入力します（例：eth0）。
MAC	MAC アドレスを入力します。
収集ルールに含まれる（In Collection Rules?）	true または false を入力します。
プロセスコマンドライン（Process Command Line）	ホストで実行されているコマンド文字列の一部を入力します（注：このファセットはインベントリフィルタの一部として保存できません）。
プロセスバイナリハッシュ（Process Binary Hash）	ホストで実行されているコマンドのプロセスハッシュを入力します（注：このファセットはインベントリフィルタの一部として保存できません）。
パッケージ情報（Package Info）	必要に応じて、パッケージ名の後にパッケージバージョンを入力します（プレフィックス # を付けます）。
パッケージ CVE（Package CVE）	CVE ID の一部または全体を入力します。
CVE スコア v2（CVE Score v2）	CVSSv2（Common Vulnerability Scoring System）スコア（数値）を入力します。

属性	説明
CVE スコア v3（CVE Score v3）	CVSSv3（Common Vulnerability Scoring System）スコア（数値）を入力します。
ユーザーラベル（User Labels）	プレフィックス が付いたユーザーラベルに由来する属性

ステップ 3 [インベントリの検索（Search Inventory）] をクリックします。結果は、[フィルタ（Filter）] フィールドの下に、4 つのタブにグループ化されて表示されます。各タブには、関連するカラムを含むテーブルが表示されます。テーブルヘッダーのファネル（じょうご）アイコンをクリックすると、追加の列を表示できます。ユーザーラベルが使用可能な場合は、プレフィックスが付き、ここで切り替えることができます。

図 *197*：インベントリの検索結果



検索結果は、次の 4 つのタブにグループ化されます。

タブ	説明
サービス	外部オーケストレータを介して検出された Kubernetes サービスおよびロードバランサを一覧表示します。このタブは、関連する外部オーケストレータが構成されている場合を除き非表示になります。
ポッド	Kubernetes ポッドを一覧表示します。このタブは、関連する外部オーケストレータが構成されている場合を除き非表示になります。
ワークロード	Secure Workload エージェントによって報告されたインベントリ項目を一覧表示します。

タブ	説明
IP アドレス	インベントリのアップロードとフローを通じて検出されたインベントリ項目を一覧表示します。

各タブの横にインベントリ数も表示されます。検索ですぐに表示される情報には、ホスト名、IP アドレス、OS、OS バージョン、サービス名、ポッド名などがあります。表示されるカラムのリストは、テーブルヘッダーのファネル（じょうご）アイコンをクリックして切り替えることができます。検索結果は、範囲ディレクトリに表示されている現在選択されている範囲に制限されます。検索結果の項目をクリックすると、それぞれのプロファイルページで詳細情報を表示できます。

各ホストの詳細は、検索結果行の IP アドレスフィールドをクリックしてアクセスできる [ワークロードプロファイル（Workload Profile）] に表示されます。詳細については、「ワークロードプロファイル」を参照してください。

サイドバーからインベントリフィルタを作成するには、トップレベルメニューから **[整理（Organize）]** > **[インベントリフィルタ（Inventory Filters）]** を選択します。[フィルタの作成（Create Filter）] ボタンをクリックします。保存するフィルタに名前を付けることができるモーダルダイアログが表示されます。

子範囲の提案

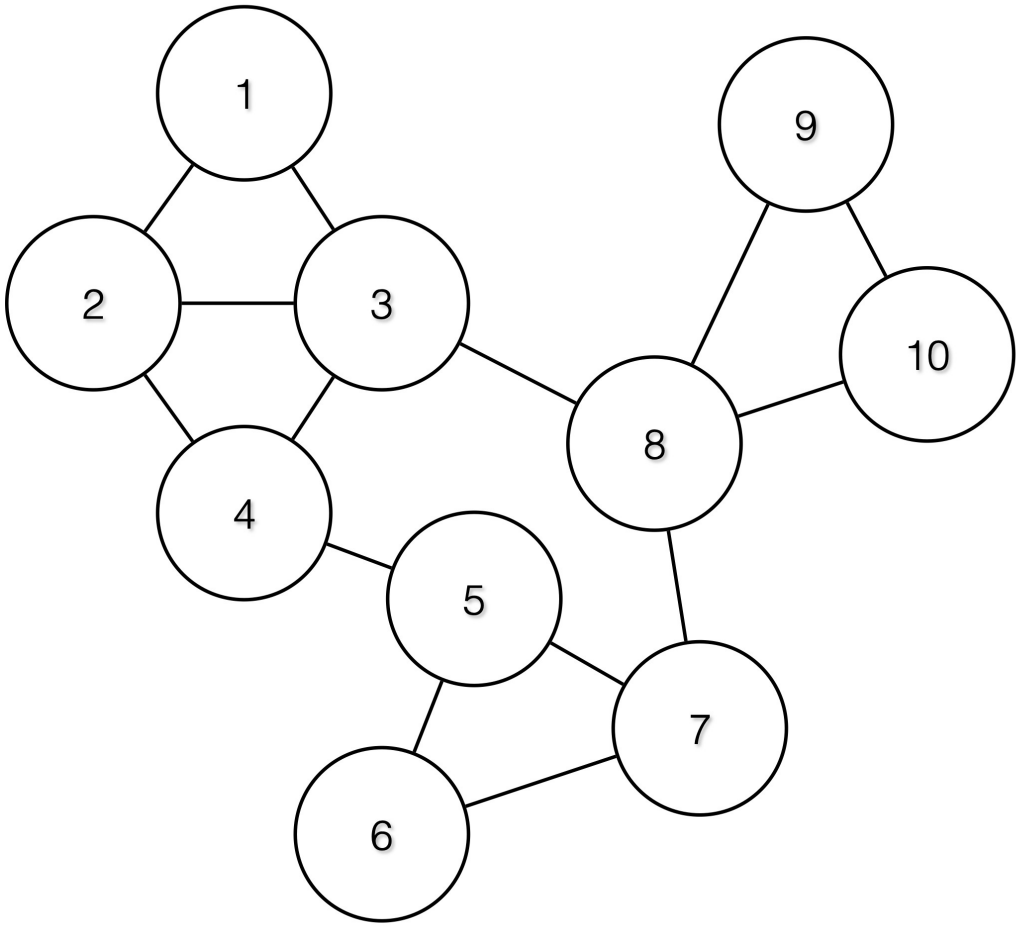
子範囲の提案は、機械学習アルゴリズム（ネットワークでのコミュニティ検出など）を使用して、範囲として機能する可能性のあるグループを検出するツールです。このツールは、範囲階層を構築するときに役立ち、特定の範囲に対してより詳細な子範囲を定義するプロセスを容易にします。候補の子範囲は提案として表示され、選択して追加できます。

概念レベルでのアルゴリズムの説明：親範囲の要求されていないメンバー間の通信に基づくグラフが最初に作成されて、グラフが処理されます（注：要求されていないメンバーとは、親の子範囲に属していないメンバーです）。たとえば、アルゴリズムでは、グラフ内の大部分を占める他のエンドポイントと通信するエンドポイントが識別されます。そのようなエンドポイントのグループが見つかった場合は、**共通サービス**グループの候補としてユーザーに表示されます。グラフの残りの部分は、**コミュニティ**として動作するグループを検出するために処理されます。コミュニティとして動作するとは、エンドポイントがグループ外のエンドポイントよりも不釣り合いに頻繁に（またはより多くのプロバイダーポートで）互いに通信することを意味します。そのような各グループは、アプリケーションまたは組織内の部門に対応している場合があります。そのようなパーティション分割により、範囲間でポリシーが希薄になる可能性もあります。

例：

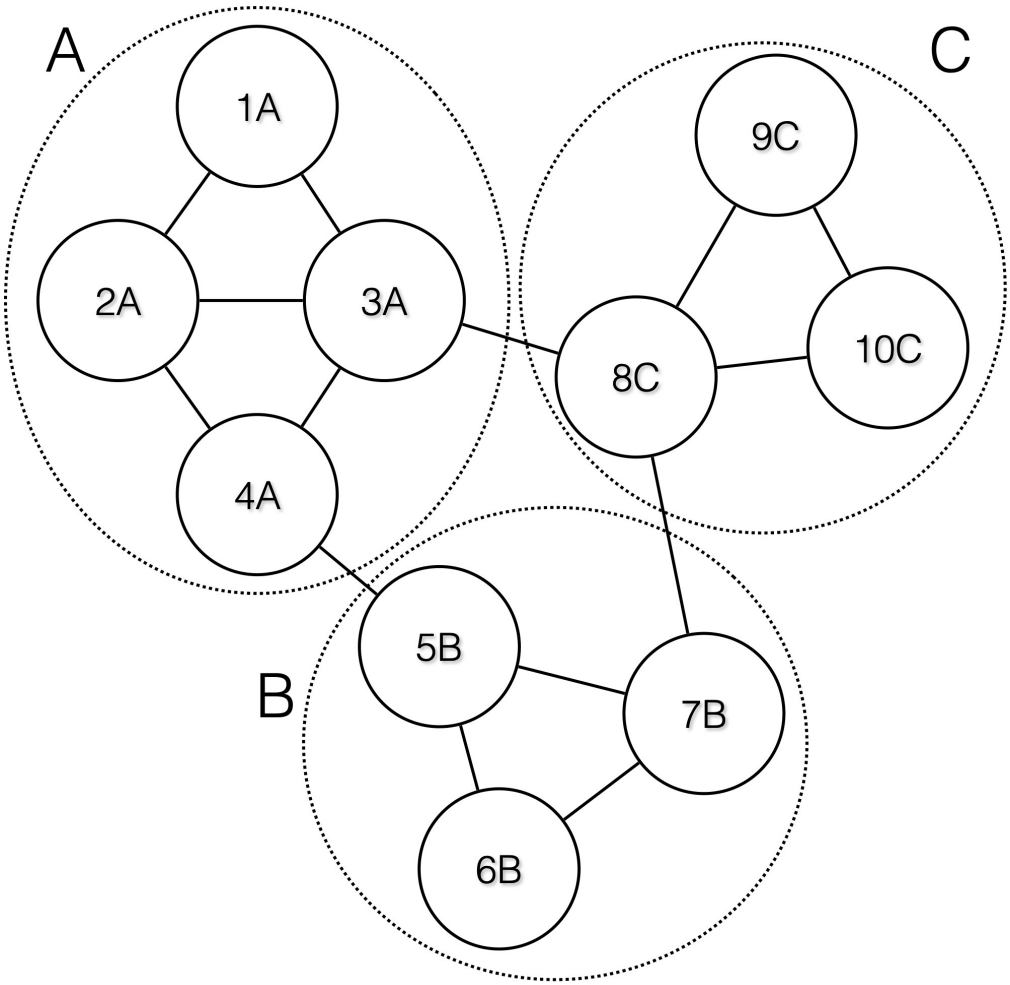
次の図の 1 から 10 を個々のエンドポイント IP とします。入力（通信）グラフは次のようになっているとします。

図 *198 :* 入力グラフ



次に、エンドポイント 1 ～ 4、5 ～ 7、および 8 ～ 10 では、相互に比較的高度な通信（エッジの数）が行われ、他のエンドポイントとの通信が比較的少ないため、一緒にグループ化されます。

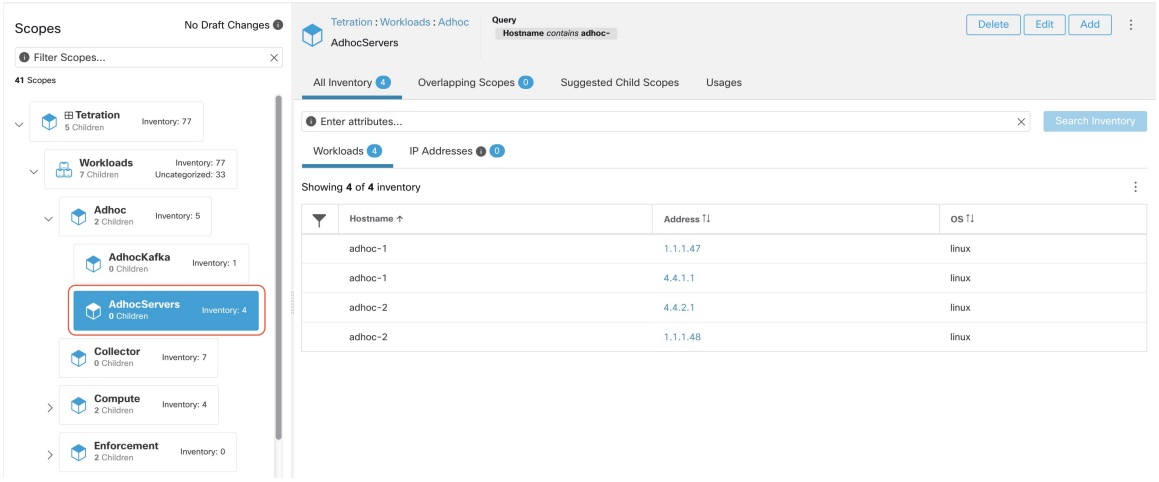
図 *199 :* 出力グループ



範囲の提案の実行手順

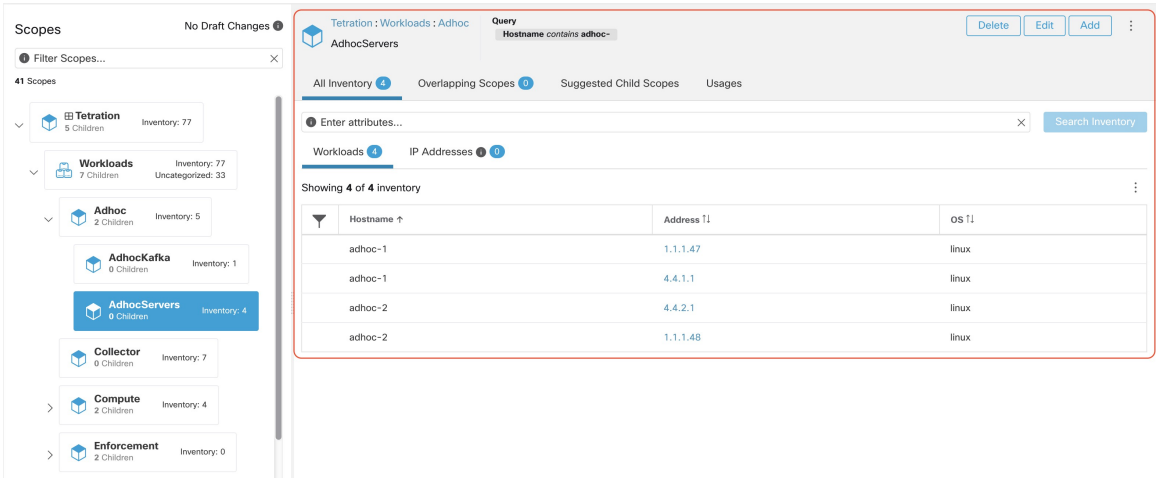
目的の範囲の範囲の提案を開始するには、範囲のページで範囲を検索して選択する必要があります。

図 *200* : 範囲の選択例



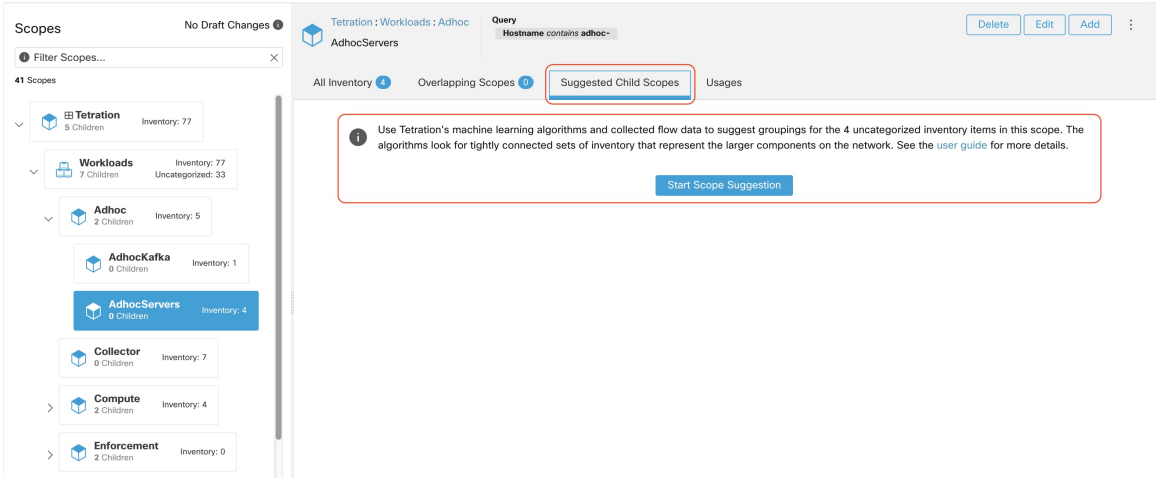
このウィンドウでは、ユーザーはインベントリ「未分類のインベントリ項目」を参照できます。つまり、現在選択されている範囲に属し、現在選択されている範囲のいずれの子範囲にも属さない項目を参照できます。**未分類のインベントリ項目**をクリックすると、このリストを表示できます。

図 *201* : 範囲ウィンドウの例



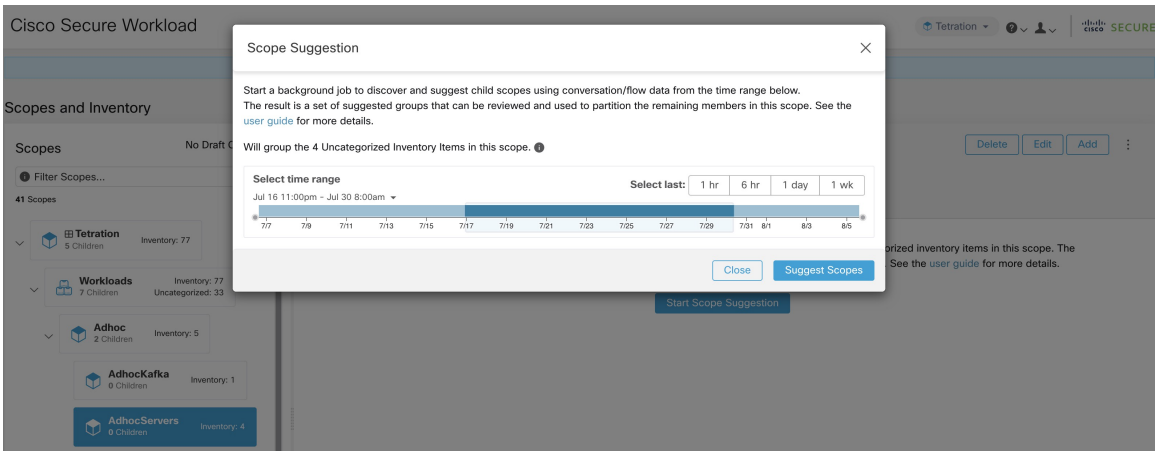
範囲を選択した後、ユーザーは[子範囲の提案（Suggest Child Scopes）]をクリックし、[範囲の提案の開始（Start Scope Suggestion）]をクリックできます（または、これが初めてでない場合は [再実行（Rerun）] をクリックします）。範囲提案の実行の入力は、分類されていないインベントリ項目になることに注意してください。

図 *202 : [*子範囲の提案（***Suggest Child Scopes***）*]* タブ



ユーザーは、範囲提案の入力として日付範囲を設定し、[範囲の提案（Suggest Scopes）] をクリックできます。範囲の提案の実行は、全体的な負荷が中程度の場合は高速であることが多く、数万回の通信で 10 から数千のエンドポイントを処理するのに数分しかかかりません。

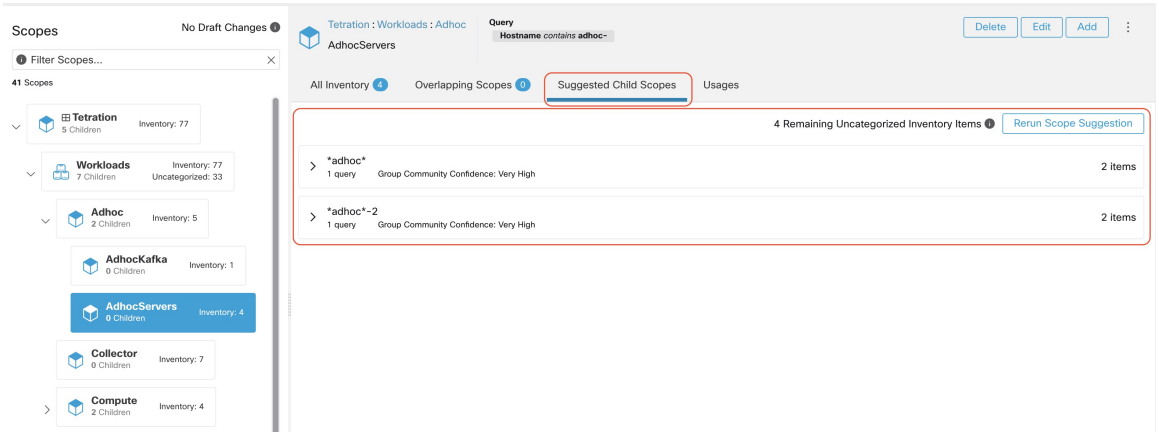
図 *203 :* 範囲提案データ範囲セレクタの例



出力は、候補のリストとしてユーザーに表示されます。現在、最大20のグループが表示され、それぞれにグループの信頼度（品質）、候補範囲名、クエリなどの情報が付随しています。検出された各グループには、関連付けられた [グループコミュニティ信頼度（Group Community Confidence）] があり、次の値のいずれかが示されます。[非常に高（Very High）]、[高（High）]、[中（Medium）]、[低（Low）]。これは、グループの [コミュニティ（Community）] プロパティの基準となります。信頼度が高いほど、エンドポイントの特定のグループのコミュニティプロパティが高くなります（グループの内側に多くのエッジがあり、外側のエッジが比較的少ない）。表示されているグループのサブセットは、[グループコミュニティ信頼度（Group Community Confidence）] に基づいて選択されています。検出されたグループは、現在、次の 4 つのグループタイプのいずれかに分類されます。

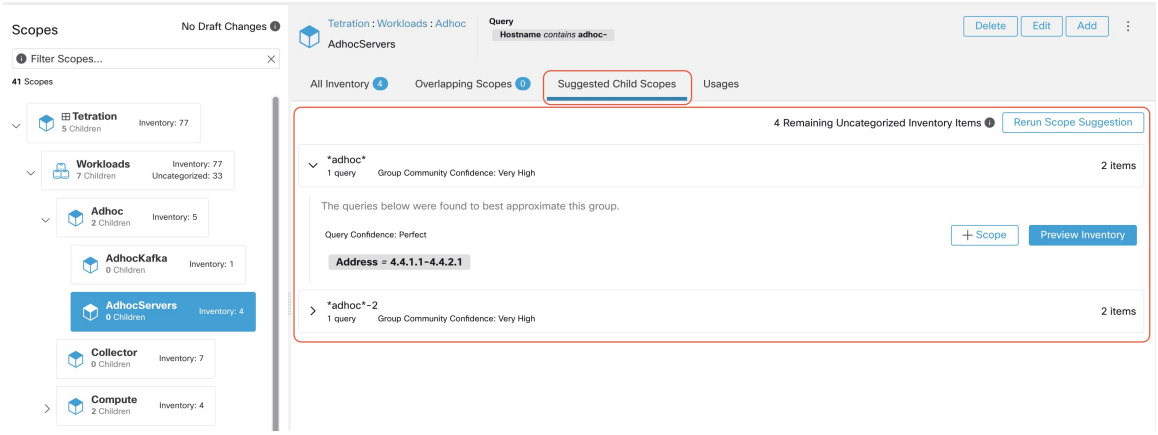
- [一般グループ（Generic Group）]：コミュニティプロパティに基づいて機械学習によって検出されたすべてのグループ。以下の特別なタイプで明示的に指定されていないグループは、一般グループであることに注意してください。
- [共通サービス（Common Service）]：このグループは、入力インベントリの多くと通信するエンドポイントで構成されます。これらのエンドポイントは、何らかの共有サービスを実行している可能性があります。
- [共通サービスクライアント（Common Service Clients）]：このグループは、[共通サービス（Common Service）] グループとのみ通信するエンドポイントで構成されます。
- [グループ化解除（Ungrouped）]：このグループは、十分な通信がないためにグループ化できないエンドポイントで構成されます。

図 *204*：範囲提案の出力例



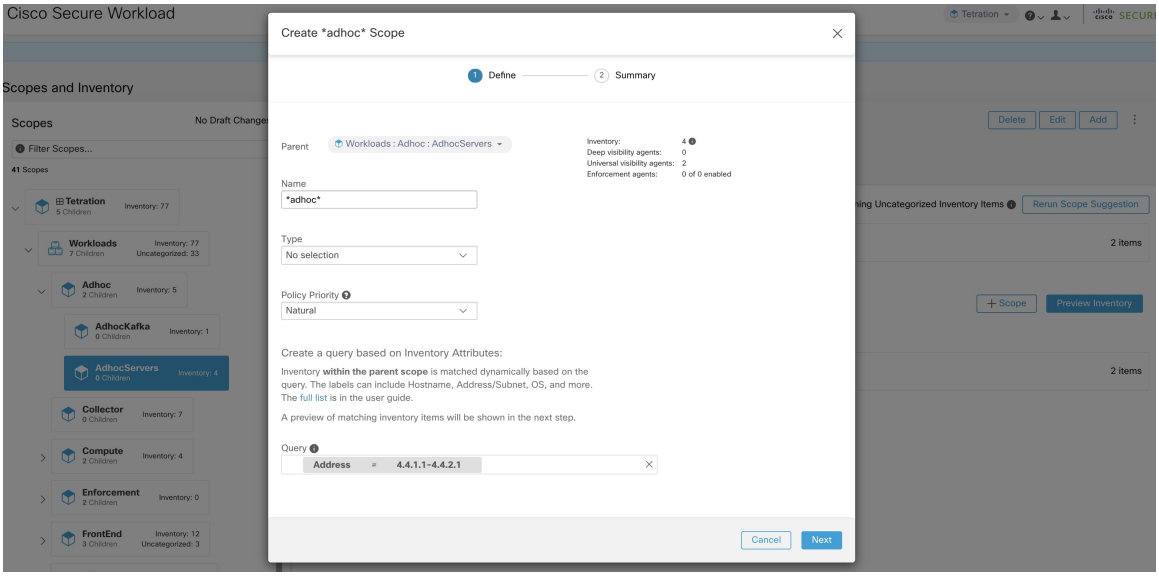
ユーザーは、検出されたグループをクリックして、選択したグループに対して生成されたクエリのリストを表示できます。ユーザーは、検出されたグループを厳密に定義するクエリの範囲と一致するインベントリをプレビューできます。クエリは、IP範囲、サブネット、ホスト名、およびユーザーがアップロードしたラベルで構成されます。各グループに関連付けられた信頼度基準があり、[クエリの信頼度（Query confidence）] と呼ばれます。次の値の範囲のいずれかとなります。 [完全（Perfect）]、[非常に高（Very High）]、[高（High）]、[中（Medium）]、[低（Low）]。クエリ生成では、まずグラフ処理と機械学習によってグループが検出され、次にグループごとにクエリが生成されます。[クエリの信頼度（Query confidence）] は、クエリの範囲がエンドポイントとどの程度一致するかの基準です。 [完全（Perfect）] のクエリ信頼度は、提案された（検出された）グループがクエリの範囲と正確に一致していることを示します。反対に、クエリの信頼度が [低（Low）] の場合は、クエリは多くの提案されたグループを取りこぼし、正確に一致できていないことを示しています。これは、クエリの範囲が[過剰なIP（Extra IPs）]となっている（検出されるグループの範囲外である）、またはクエリに多くの[欠落IP（Missing IPs）] がある（クエリの範囲外である）ことを意味します。

図 *205 :* 範囲提案出力クエリの例



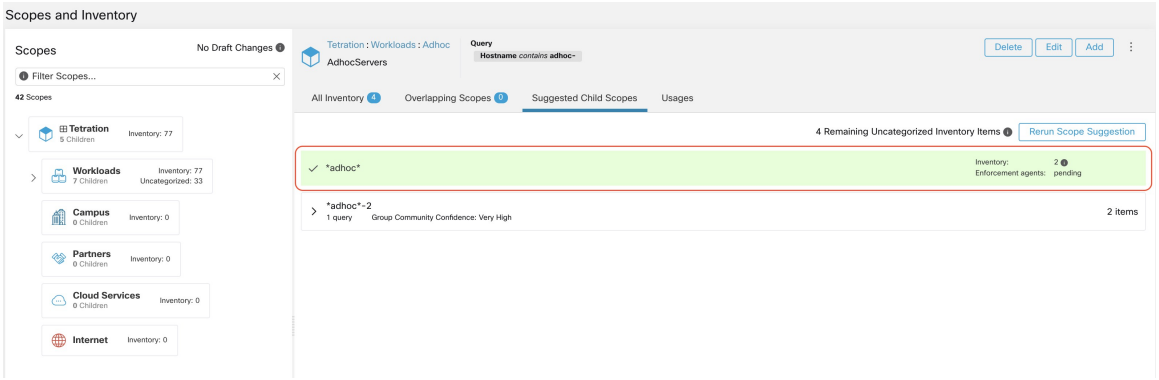
ユーザーは、[+範囲（+ Scope）] ボタンをクリックして、グループ名とグループクエリを編集できる編集ウィンドウに移動できます。ユーザーは、クエリおよび一致する IP を調べ、クエリを調整して一部の IP を追加または削除する必要があるかどうかを判断できます。問題がなければ、ユーザーは [次へ（Next）] をクリックして確認し、ドラフトビューキャンバスでグループを範囲に変換できます。

図 *206 :* 範囲提案編集画面の例



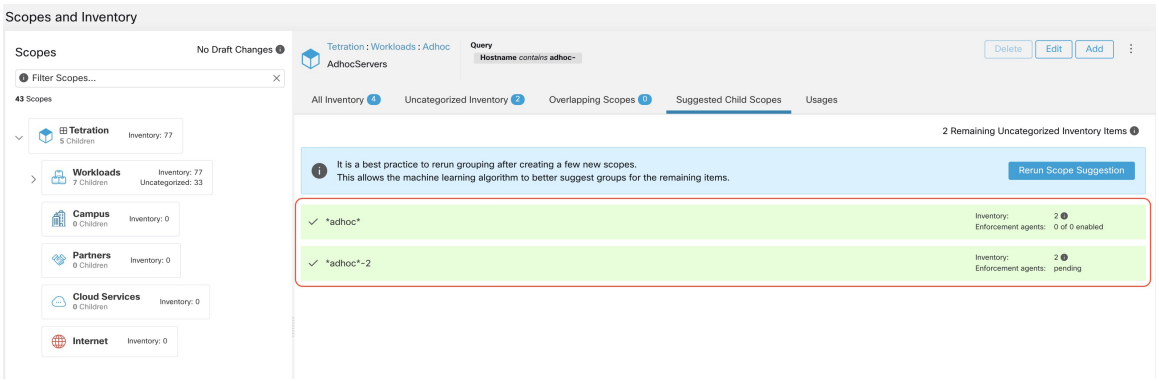
提案されたグループを範囲に変換すると、グループスロットが緑色になり、[未分類のインベントリ項目（Uncategorized Inventory Item）] の数が減少します。

図 *207 : 1* つの提案されたグループを範囲に変換した後の範囲提案出力の例



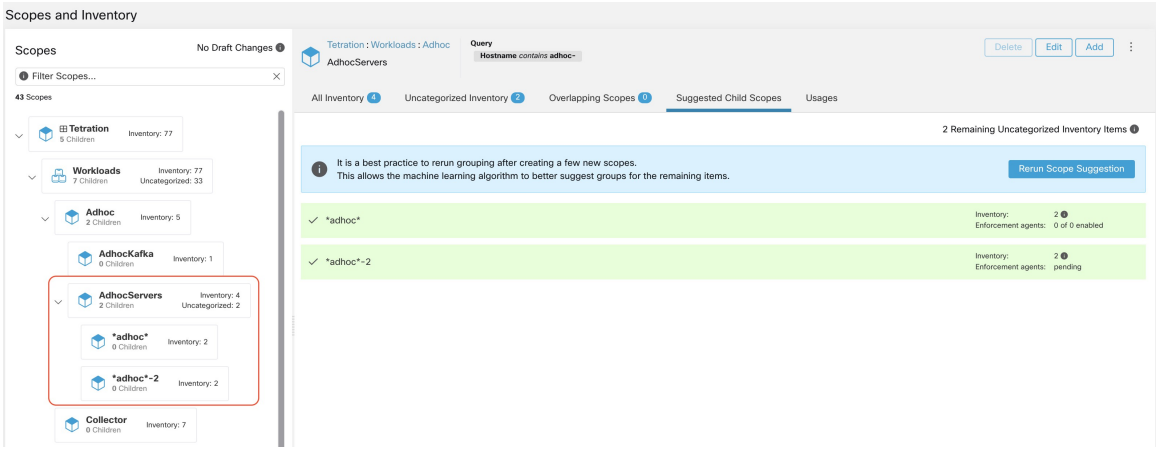
ユーザーは、グループの残りのリストから範囲作成のプロセスを繰り返すことができます。推奨されるワークフローは、1つ以上の範囲を作成してから、**範囲の提案**を再実行することです。[未分類のインベントリ項目（Uncategorized Inventory Item）] の数がゼロの場合は、（現在選択されている親範囲について）さらに範囲を限定するインベントリが残っていないことを示します。

図 *208 :* 複数の範囲作成後の範囲提案出力の例



範囲の作成プロセスが完了したら（未分類の数は 0）、ユーザーは、新しく作成された子範囲でこのプロセスを繰り返して、必要に応じてより深い範囲ツリーを生成できます。

図 *209 :* 最初の範囲提案と作成後の範囲リストの例



（注） 範囲内の未分類の項目がうまく分類されない可能性もあります（例：コミュニティを形成しない）。その場合、アルゴリズムはグループ化を返さないことがあります（空の結果）。

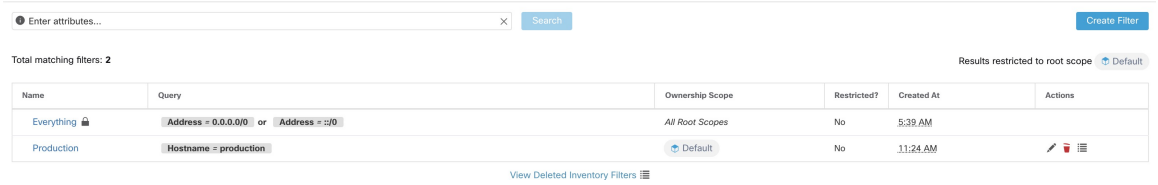
フィルタ

フィルタは、ポリシー、設定インテントなどを定義するときに使用できる、保存済みインベントリ検索です。各フィルタは、フィルタの所有範囲として定義されている範囲に関連付ける必要があります。

既存のフィルタを表示するには、左側のナビゲーションメニューから**[整理（Organize）]>[インベントリフィルタ（Inventory Filters）]**を選択します。また、該当する範囲の任意のワークスペースに含まれる任意の範囲に固有のインベントリフィルタを表示することもできます。

フィルタのリストは、現在選択されている範囲のルートに基づいて制限されます。

図 *210 :* インベントリ フィルタ



選択した親範囲に関するインベントリメンバーシップの変更を確認するには、範囲/フィルタ変更の影響を確認 ウィンドウにアクセスします。

インベントリフィルタの作成

さまざまな目的でインベントリフィルタを作成できます。たとえば、インベントリフィルタを使用して次のことができます。

- 範囲内のワークロードのサブセットに固有のポリシーを作成または検出します。

  たとえば、APIインターフェイスを介してのみアクセスされるアプリケーションがある場合、範囲内にAPIサーバーのグループを作成して、そのトラフィックを許可し、そのアプリケーションの他のすべてのワークロードへのアクセスをブロックするポリシーを作成できます。

- 多くの範囲にまたがって存在する可能性のあるワークロードのポリシーを作成します。

  たとえば、特定のオペレーティングシステムを実行しているネットワーク上のすべてのワークロードに適用されるポリシーを作成する必要がある場合は、複数（またはすべて）の範囲にまたがるインベントリフィルタを作成できます。



ヒント　既存のクラスタをインベントリフィルタに変換するには、「クラスタをインベントリフィルタに変換する （601 ページ）」を参照してください。

手順

ステップ 1　次のいずれかの場所に移動します。

- [整理（Organize）] > [インベントリフィルタ（Inventory Filters）] を選択します。
- インベントリフィルタを作成する範囲内の任意のワークスペースに移動し、[ポリシーの管理（Manage Policies）]、[フィルタ（Filters）]、[インベントリフィルタ（Inventory Filters）] の順にクリックします

ステップ 2　[フィルタの作成（Create Filter）] または [インベントリフィルタの追加（Add Inventory Filter）] をクリックします。

ステップ 3　名前、説明、およびフィルタに含めるワークロードのみを指定したクエリを追加します。

ステップ 4　[詳細オプションを表示（Show advanced options）] が表示されている場合は、このリンクをクリックします。

ステップ 5　このフィルタの範囲を指定します。

選択した範囲によって、次のことが決定されます。

- このフィルタを変更できるユーザー:

  管理者がこのフィルタを変更するには、指定された範囲またはその先祖のいずれかへの書き込みアクセス権を持っている必要があります。

- フィルタに含まれるワークロード（この手順の他の設定に依存）。

ステップ 6 設定オプションは次のとおりです。

目的	操作手順
このフィルタで指定された範囲のメンバーであるかどうかに関係なく、フィルタのクエリ条件を満たすワークロードを含めます。	[クエリを所有権の範囲に制限する（Restrict query to ownership scope）] の選択を解除します。
このフィルタで指定された範囲のメンバーであるワークロードのみを含めます。	[クエリを所有権の範囲に制限する（Restrict query to ownership scope）] を選択します。
自動ポリシー検出を許可して、このフィルタで定義された一連のワークロードに固有のポリシーを提案します。 これらのワークロードは、フィルタで指定された範囲のサブセットである必要があります。	[クエリを所有権の範囲に制限する（Restrict query to ownership scope）] と [範囲外のサービスを提供する（Provides a service external of its scope）] の両方を選択します。 後者を選択するには、前者を選択する必要があります。 このフィルタを使用するには、外部依存関係を設定する必要があります。ワークスペースの外部依存関係の微調整（551 ページ）を参照してください。

ステップ 7 [次へ（Next）] をクリックします。

ステップ 8 詳細を確認して [作成（Create）] をクリックします。

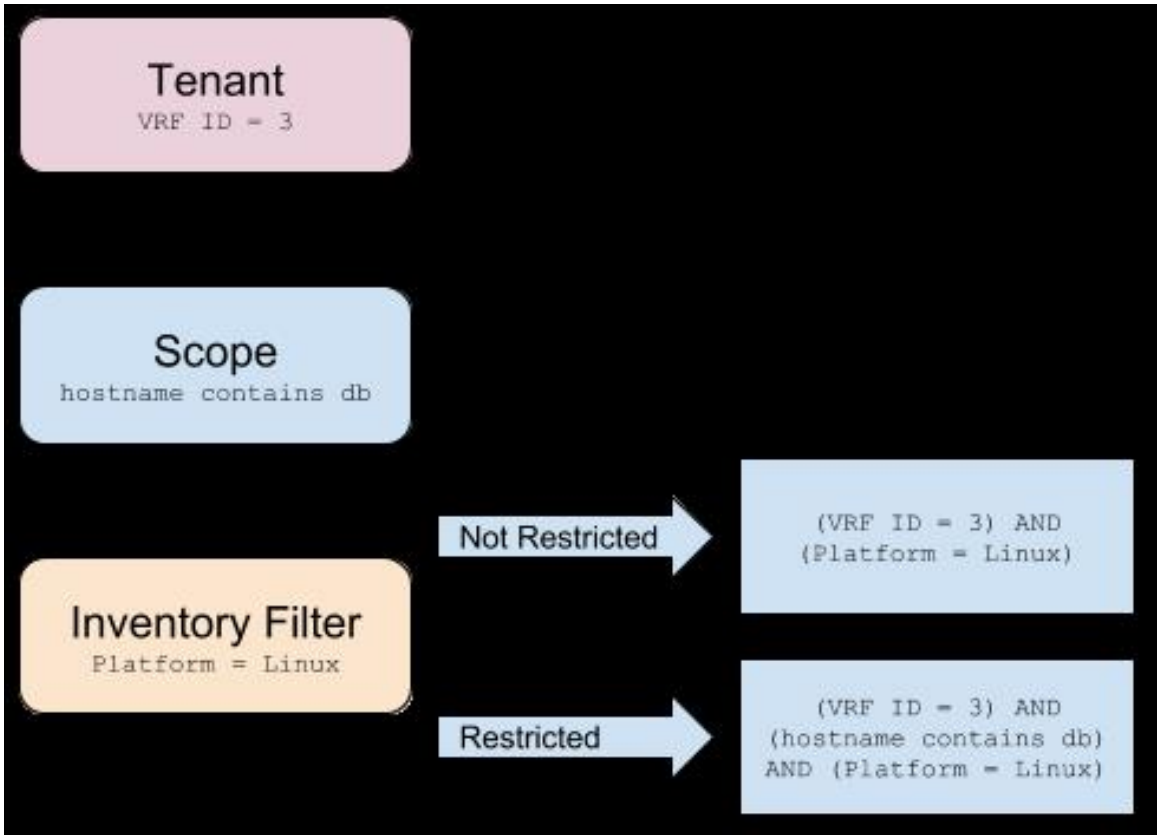
所有権の範囲に制限

範囲がフィルタによって一致するインベントリに影響を与えるかどうかは、[所有権の範囲に制限（Restrict to Ownership Scope?）] チェックボックスによって決まります。

たとえば、次のようなクエリ構造があるとします。

1. クエリで指定されたテナント：`VRF ID = 3`
2. クエリで指定されたこのテナント内の範囲：`hostname contains db`
3. クエリで指定されたこの範囲に割り当てられているインベントリフィルタ：`Platform = Linux`

図 *211* : テナント、範囲、インベントリフィルタ構造



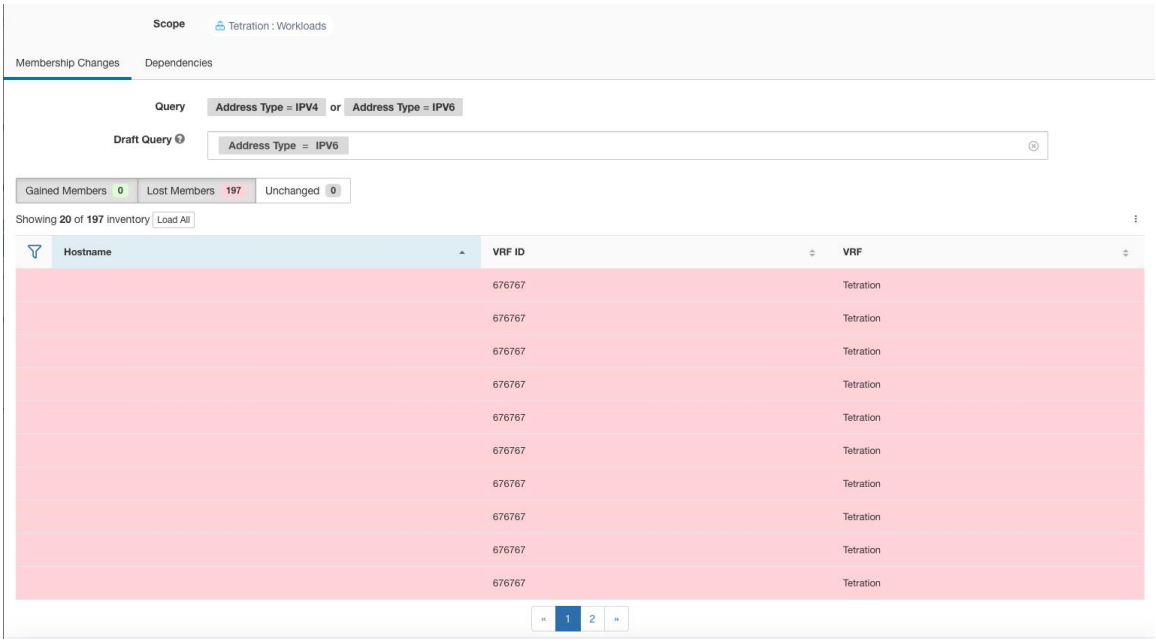
- [所有権の範囲に制限（Restrict to Ownership Scope）] チェックボックスがオフになっている場合、このフィルタは、インベントリフィルタにも一致するテナント内のすべてのホストに一致します。有効なクエリは `(VRF ID = 3) AND (Platform = Linux)` になります。
- [所有権の範囲に制限（Restrict to Ownership Scope）] チェックボックスがオンになっている場合、このフィルタは、インベントリフィルタにも一致するテナントと範囲内のホストのみに一致します。有効なクエリは `(VRF ID = 3) AND (hostname contains db) AND (Platform = Linux)` になります。

範囲/フィルタ変更の影響を確認

範囲クエリを更新すると、コミットされた後の範囲のインベントリメンバーシップに影響を与える可能性があります。同様に、直接保存されるフィルタクエリの変更も、範囲のインベントリメンバーシップに影響を与える可能性があります。[範囲（Scope）] または [フィルタ編集（Filter Edit)] モーダルのいずれかで [クエリ変更の影響の確認（Review query change impact）] リンクをたどることで、新しいクエリと古いクエリの間のメンバーシップの変更を確認できます。さらに、範囲またはフィルタの依存関係を把握すると、影響の分析に役立つだけでなく、

範囲の削除を妨げる必要なすべてのオブジェクトを削除する際にも役立ちます。詳細については、[依存関係（Dependencies）] タブにもアクセスして、範囲の依存関係ツリーを検討してください。

図 *212 :* メンバーシップテーブルのダウンロード



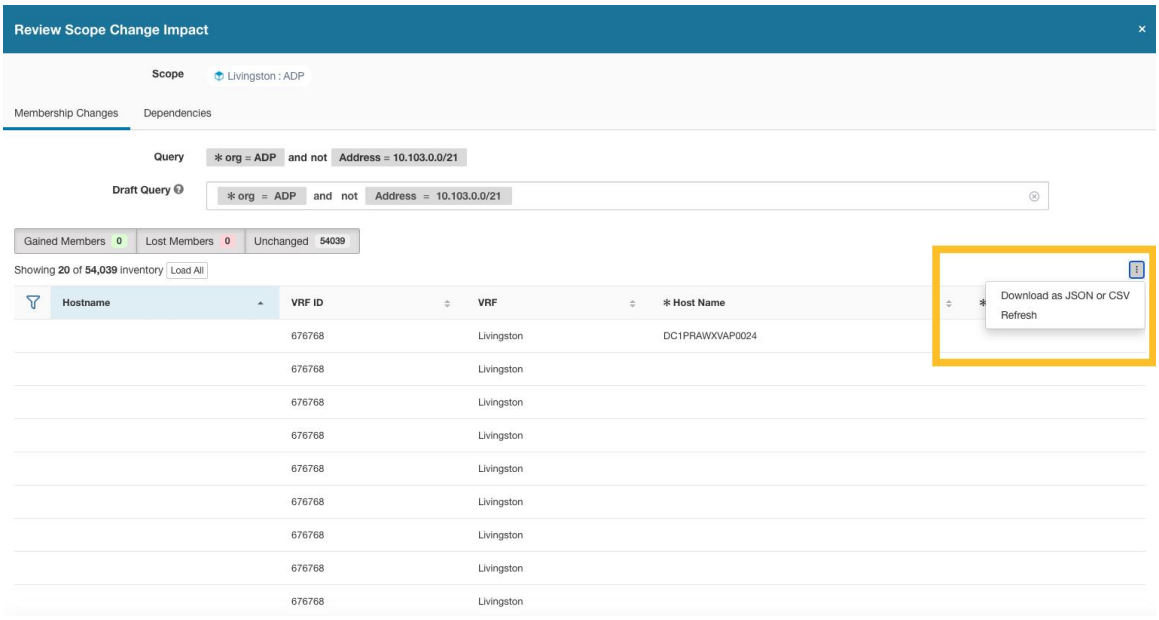
範囲クエリ変更影響モーダル

[メンバーシップの変更（Membership Changes）] タブと [依存関係（Dependencies）] タブの両方にアクセスするには、[範囲編集（Scope Edit）] ウィンドウで [クエリ変更の影響の確認（Review query change impact）] リンクをクリックします。

メンバーシップの変更

メンバーシップビューのインベントリテーブルには、デフォルトですべての列が表示されます。表示する行は選択できます。さらに、インベントリが[ゲイン（Gained）]、[喪失（Lost）]、[変更なし（Unchanged）] のいずれかであるかを識別する追加の Diff 列を含む、選択したメンバーシップ列と行の csv または json をダウンロードできます。ダウンロードに必要なすべてのテーブルの選択がテーブルビューに表示されていることを確認してください。

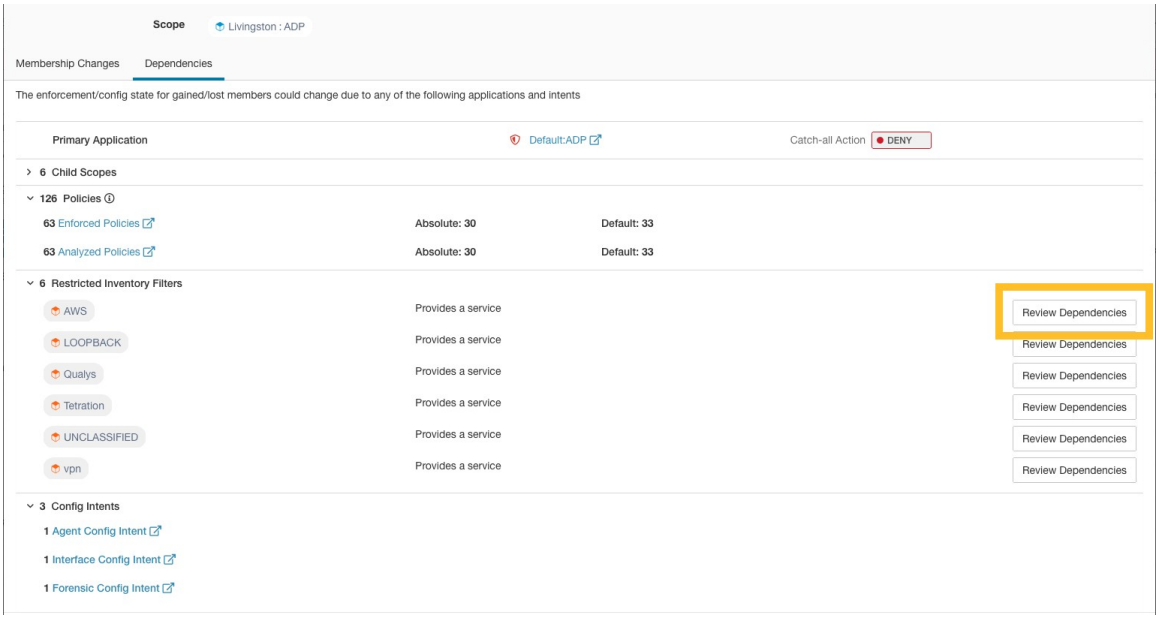
図 *213* : 範囲メンバーシップの変更



依存関係

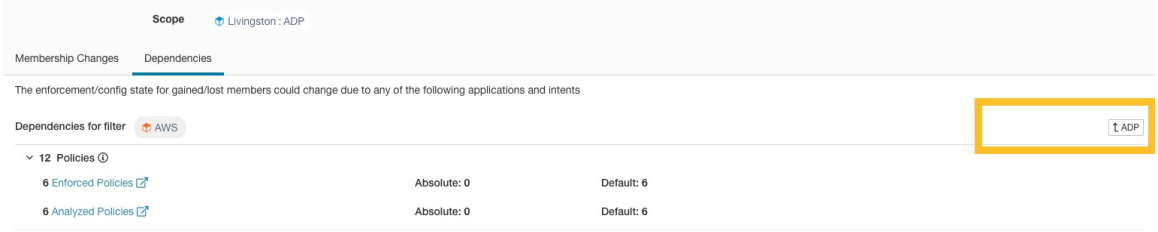
[依存関係の確認（Review Dependencies）] をさらに選択することで、ネストされた依存関係までトラバースできます。

図 *214* : 依存関係の確認



該当する親リンクを選択することで、依存関係ツリーをトラバースできます。

図 *215 :* 親リンク



存在する可能性のある範囲の依存関係は次のとおりです。

表 *24 :* 存在する可能性のある範囲の依存関係は次のとおりです

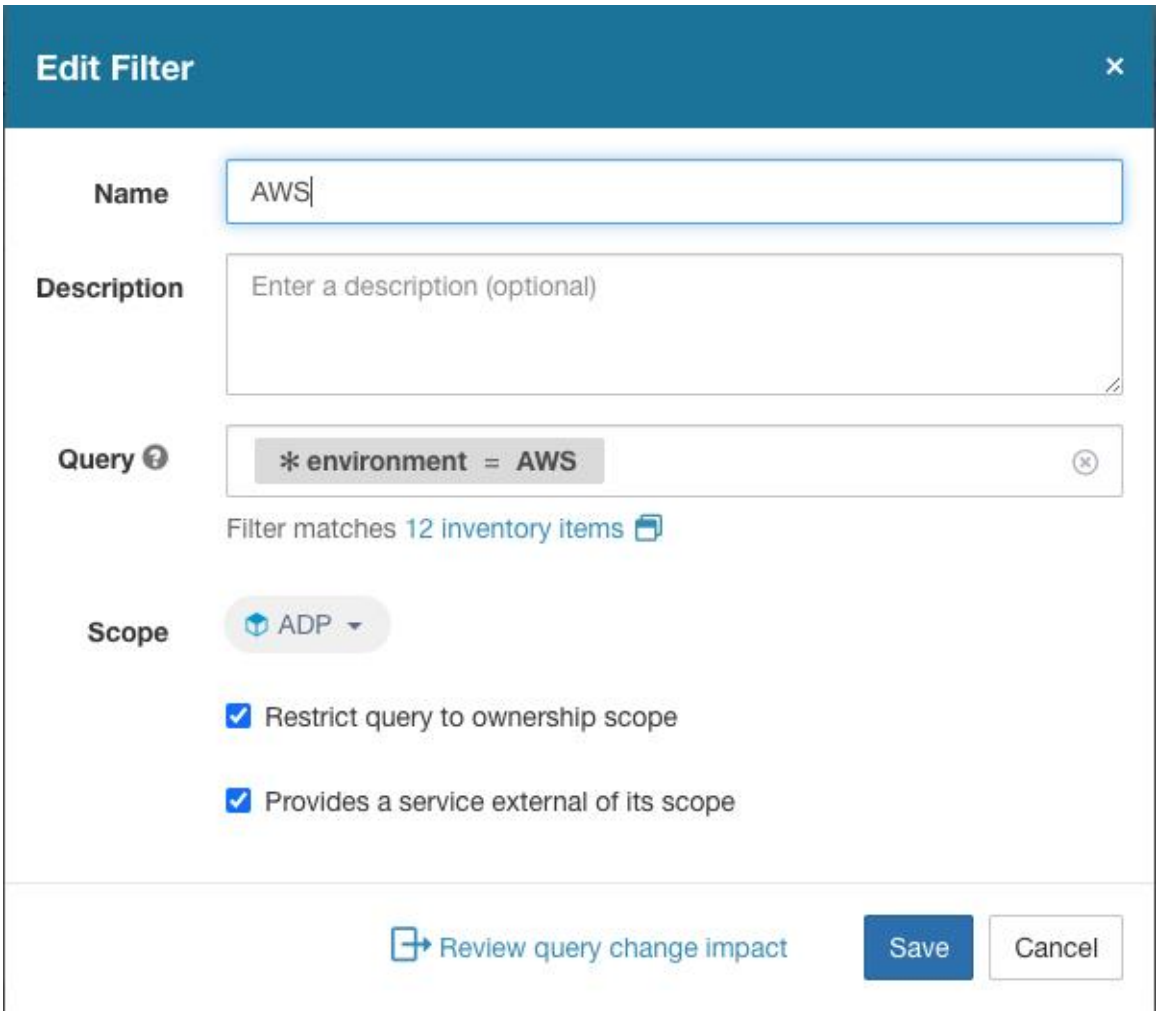
タイプ	説明
アプリケーション	プライマリおよびセカンダリアプリケーション名と、セグメンテーションの下にある特定のワークスペースへのリンクがあります。
子範囲	子範囲の名前と子範囲の詳細ビューへのリンクがあります。下位レベルの依存関係にドリルダウンできます。
ポリシー	分析および適用されたポリシーの数と、選択した範囲でフィルタ処理されたそれぞれのグローバルポリシービューへのリンクがあります。
制限付きインベントリフィルタ	フィルタの名前と子フィルタの詳細ビューへのリンクがあります。下位レベルの依存関係にドリルダウンできます。
構成インテント	構成インテントの名前、およびエージェント、インターフェイス、フォレンジックの構成インテントビューへのリンクがあります

フィルタクエリ変更影響モーダル

[メンバーシップの変更（Membership Changes）] タブと [依存関係（Dependencies）] タブの両方にアクセスするには、[インベントリフィルタの編集（Inventory Filter Edit）] ウィンドウで [クエリ変更の影響の確認（Review query change impact）] リンクをクリックします。

メンバーシップの変更

図 *216* : インベントリフィルタのメンバーシップの変更



依存関係

存在する可能性のあるフィルタの依存関係は次のとおりです。

タイプ	説明
ポリシー	分析および適用されたポリシーの数と、選択した範囲でフィルタ処理されたそれぞれのグローバルポリシービューへのリンク
[設定インテント（Config Intents）]	エージェント、インターフェイス、フォレンジック設定インテントビューへの名前およびリンクがあります

インベントリプロファイル

（注） インベントリのプロファイルページは、さまざまな場所からリンクされています。インベントリプロファイルを表示する方法の1つは、インベントリの検索を実行し、IPアドレスをクリックしてそのプロファイルに移動することです。[範囲とインベントリ（Scopes and Inventory）] ページで作業している場合は、[ワークロード（Workloads）] タブの IP アドレスではなく、[IP アドレス（IP addresses）] タブの IP アドレスをクリックします（[ワークロード（Workloads）] タブで IP アドレスをクリックすると、インベントリプロファイルではなく、ワークロードプロファイルが表示されます）。

インベントリについては、次の情報が表示されます。

フィールド	説明
[範囲（Scopes）]	インベントリが属する範囲のリスト。
[インベントリタイプ（Inventory Type）]	• [フロー学習（Flow Learnt）] インベントリは、観測されたフローに基づいて登録されました。 • [ラベル付き（Labeled）] のインベントリは、インベントリアップロードユーティリティを使用して手動でアップロードされました。 • [エージェント（Agent）] インベントリは、ホストにインストールされているソフトウェアエージェントによって報告されました。 • [タグ（Tagged）] インベントリは、コネクタまたは外部オーケストレータによって報告されました。
[ユーザーラベル（User Labels）]	このインベントリにユーザーがアップロードした属性のリスト。詳細については、「ワークロードラベル」を参照してください。

追加情報は、次の両方に該当する場合にのみ表示されます。

1. インベントリがクラウドコネクタを介して取り込まれている。
2. インベントリが存在する仮想ネットワークに対してセグメンテーションが有効になっている。

フィールド	説明
[適用の状態（Enforcement Health）]	ホスト ソフトウェア エージェントのステータス情報。詳細については、「[エージェントの正常性（Agent Health）] タブ」を参照してください。
[具体的なポリシー（Concrete Polices）]	このタブには、ホストに適用される Secure Workload の具体的な適用ポリシーが表示されます。詳細については、「[具体的なポリシー（Concrete Policies）] タブ」を参照してください。
[セキュリティグループ（Security Groups）]	このインベントリに適用されるセキュリティグループとそのポリシーのリスト。

インベントリプロファイル情報

フィールド	説明
[試験的グループ（Experimental Groups）]	ポリシーのライブ分析に使用されるクラスタまたはユーザー定義のインベントリフィルタのリスト。
[適用グループ（Enforcement Groups）]	ポリシーの適用に使用されるクラスタまたはユーザー定義のインベントリフィルタのリスト。分析対象のポリシーやシステムで適用されているポリシーのバージョンによっては、適用グループが試験的グループとは異なる場合があります。

（注） 次の場合、IP アドレスのインベントリプロファイルの詳細が表示されない場合があります。

- インベントリが収集ルールから除外されている場合。
- 単方向フローではインベントリが 2 分間だけ使用できるようになり、その後に削除されます。
- 双方向フローでは、インベントリが 30 日間使用できますが、この 30 日間にこれ以上フローが観察されない場合、インベントリの詳細は削除されます。

ワークロード プロファイル

ワークロードプロファイルには、Secure Workload ソフトウェアエージェントがインストールされているホストに関する詳細情報が表示されます。ここでは、ワークロードプロファイルとそれに含まれる情報を表示する方法について説明します。

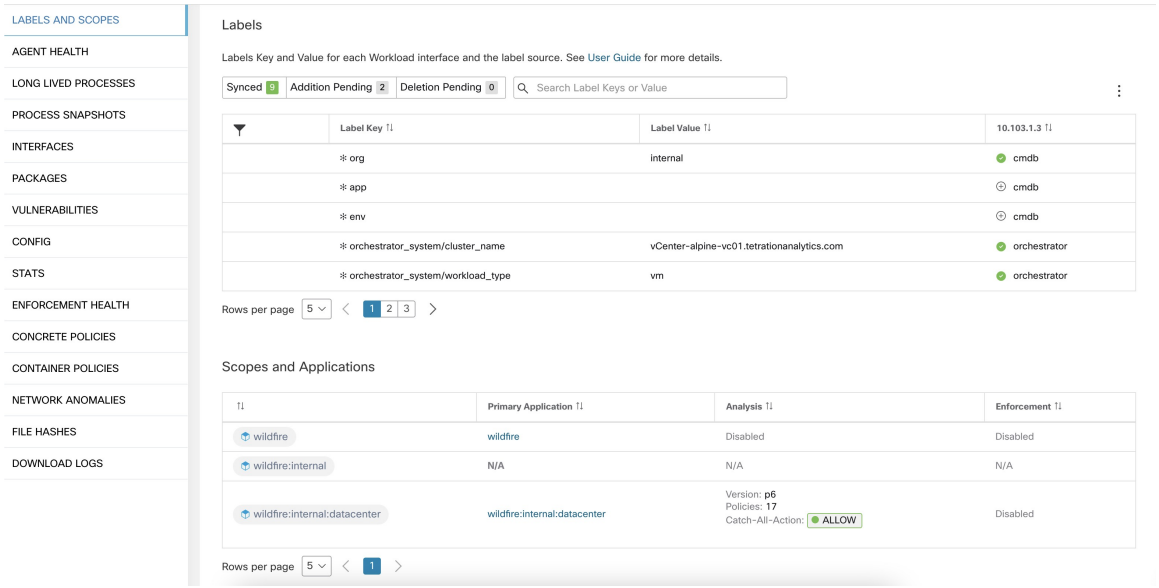
（注） ワークロードのプロファイルページは、さまざまな場所からリンクされています。ワークロードプロファイルを表示する方法の 1 つは、検索で説明したように、ホストの検索を実行することです。

インベントリ検索の結果から、ホストの IP アドレスをクリックして、そのプロファイルに移動します。ホストにインストールされているエージェントのタイプに基づいて、次のタブがページに表示されます。このインベントリが属するホストに Secure Workload ソフトウェアエージェントがインストールされていない場合、インベントリ プロファイル ページが表示される可能性があることに注意してください。

ラベルと範囲タブ

このタブには、適用グループと試験的グループ、ホストが属する範囲が含まれています。試験的グループは、ポリシーのライブ分析に使用するインベントリフィルタであり、適用グループはポリシーの適用に使用するフィルタです。これらのグループは、分析対象のポリシーやシステムで適用されているポリシーのバージョンに応じて異なる場合があります。

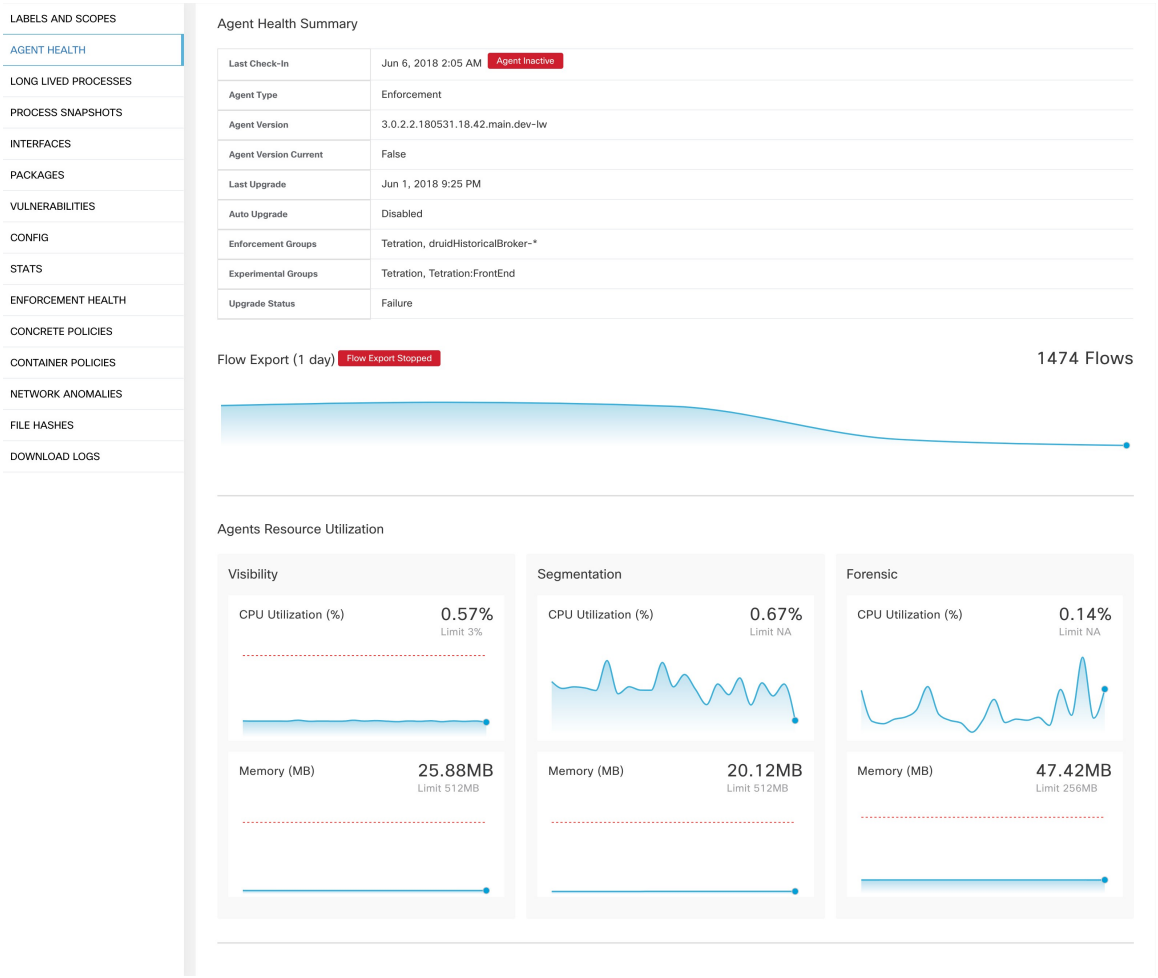
図 *217 :* ワークロードのラベルと範囲



[エージェントの正常性（Agent Health）] タブ

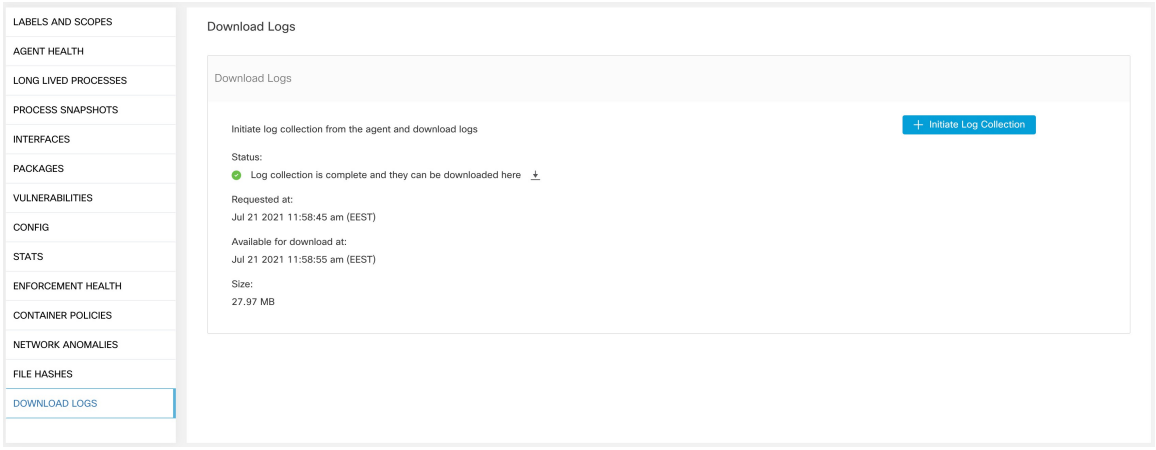
タイプ、OS プラットフォーム、エージェントのバージョン、最終チェックイン時刻などのホスト ソフトウェア エージェントのステータス情報も、[エージェントの正常性（Agent Health）] タブに表示されます。詳細については、「ソフトウェアエージェントの設定」を参照してください。このタブには、1 日あたりに発生したトラフィックのバイト数とパケット数に関する詳細な時系列データも表示されます。

図 *218* : ワークロードエージェントの正常性の詳細



ルート範囲の所有者特権を持つユーザーの場合、概要ページには、そのルート範囲内の優れた可視性および適用エージェント（バージョン 3.3 以降）のためにエージェントログを収集およびダウンロードするセクションも含まれます。また、この機能は、プラットフォーム AIX および SUSE Linux Enterprise Server（IBM Z アーキテクチャ上の s390x-Linux）で実行されているエージェントでは使用できないことに注意してください。[ログ収集の開始（Initiate Log Collection）] ボタンを使用してエージェントからログを収集すると、数分でログをダウンロードできるようになります。ダウンロードに失敗した場合は、ログの収集を再試行してから、もう一度ダウンロードを試行してください。

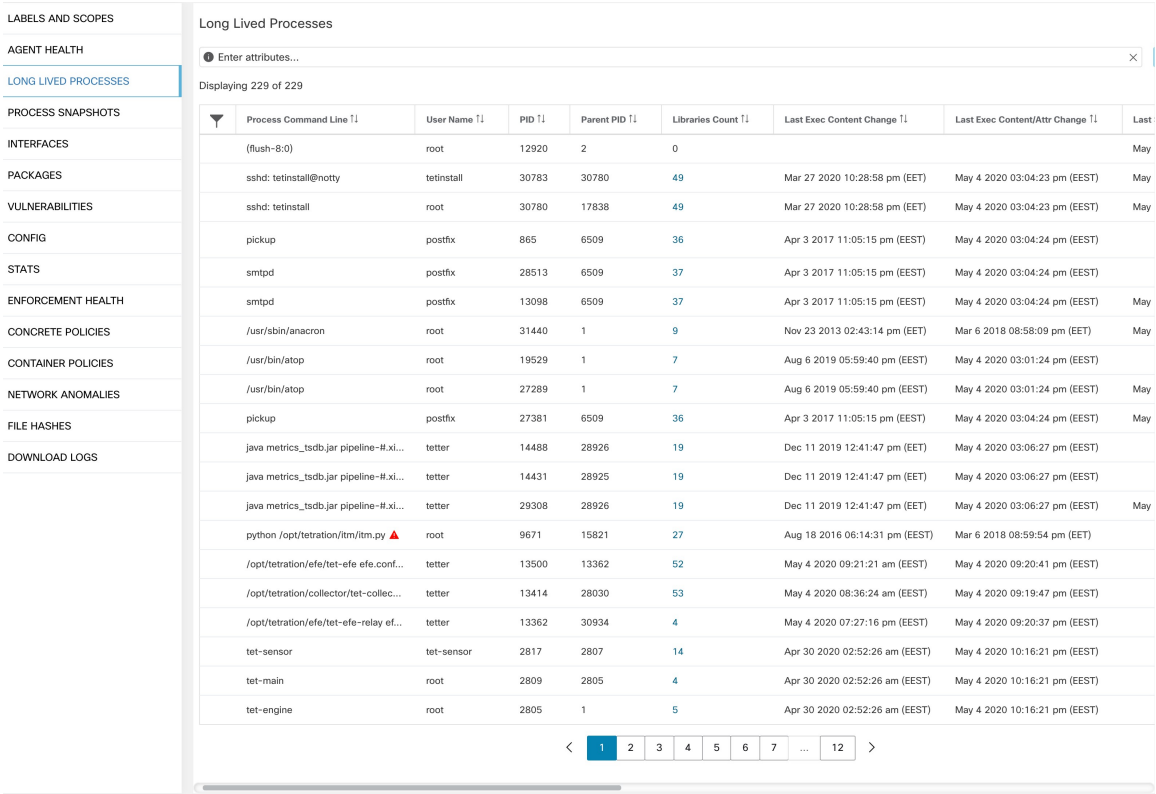
図 *219 : Agent Logs*



[プロセスリスト（Process List）] タブ

このタブには、ホストで実行されているプロセスのリストが表示されます。フィルタを使用して、以下の表ヘッダーに示されているプロセスの属性に基づいて、プロセスのリストを絞り込むこともできます。

図 *220 :* ワークロードプロセスリスト



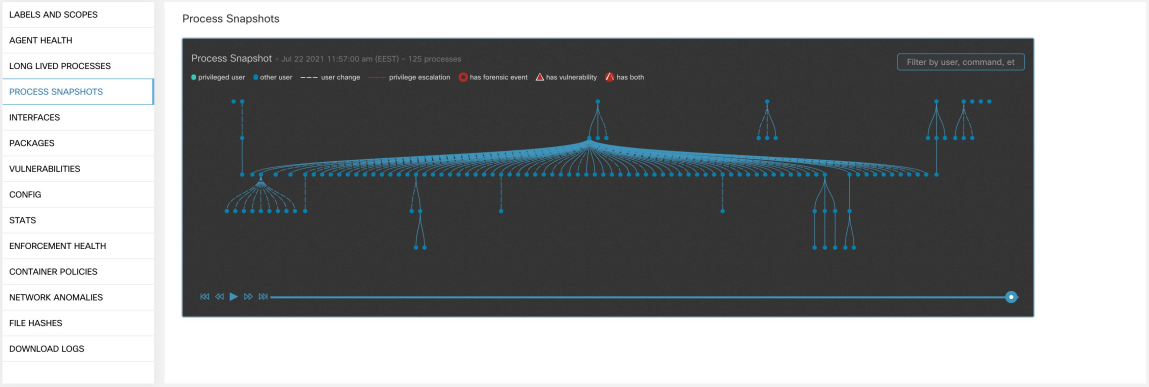
属性の説明：

属性	説明
最後の実行コンテンツの変更（Last Exec Content Change）	Linux の mtime に似ています。ファイルの内容のみが変更されたときのタイムスタンプです。
最後の実行コンテンツの変更（Last Exec Content Change）	Linux の ctime に似ています。ファイルの内容または属性が変更されたときのタイムスタンプです。
Last Seen	プロセスが最後に観察された時刻。プロセスが停止したときに使用できます。
CPU 使用率	過去 1 時間のプロセスによる CPU 使用率の傾向。
メモリ 使用法	過去 1 時間のプロセスによるメモリ使用量の傾向。
プロセスバイナリハッシュ（Process Binary Hash）	プロセスバイナリの 16 進文字列の SHA256 ハッシュ。略してプロセスハッシュとも呼ばれます。カーネルプロセスでは使用できません。
異常スコア（Anomaly Score）	プロセスハッシュ（異常）スコア。詳細については、「プロセスハッシュの異常検出」を参照してください。
判定	プロセスハッシュの判定（悪意があるまたは無害のいずれか）。判定は、プロセスハッシュがユーザー定義のハッシュリストまたは既知の脅威インテリジェンス ハッシュ データベースに属しているかどうかに基づいて決定されます。詳細については、「プロセスハッシュの異常検出」を参照してください。
判定ソース（Verdict Source）	判定のソース。判定ソースは、ユーザー定義、Secure Workload クラウド、または NIST のいずれかです。この属性は、以前のリリースではハッシュ DB ソースと呼ばれていました。詳細については、「プロセスハッシュの異常検出」を参照してください。

[プロセススナップショット（Process Snapshot）] タブ

このタブには、ワークロードで観察された検索可能なプロセスツリーが表示されます。

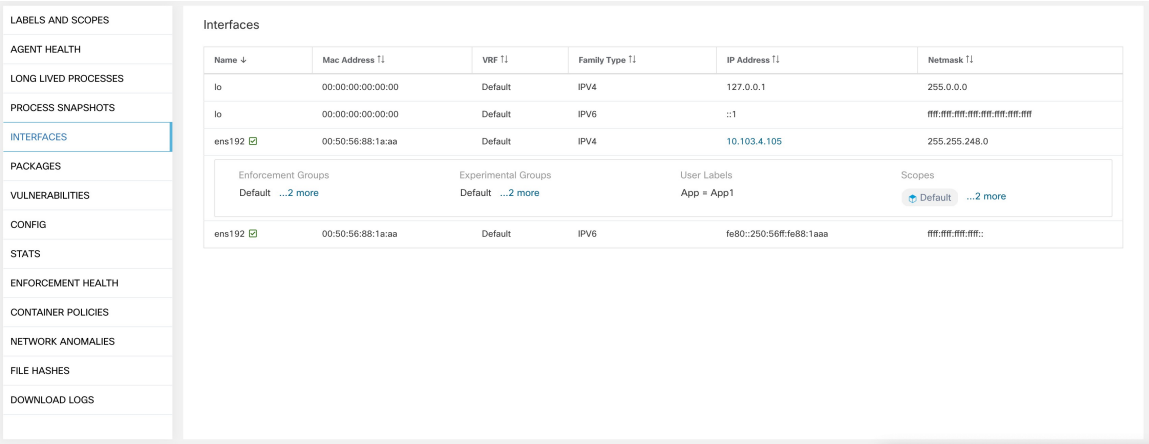
図 *221 :* ワークロード プロセス スナップショット



[Interfaces] タブ

このタブには、ホストにインストールされているネットワーク インターフェイスに関する詳細が表示されます。すべてのタイプのソフトウェアエージェントで使用できます。

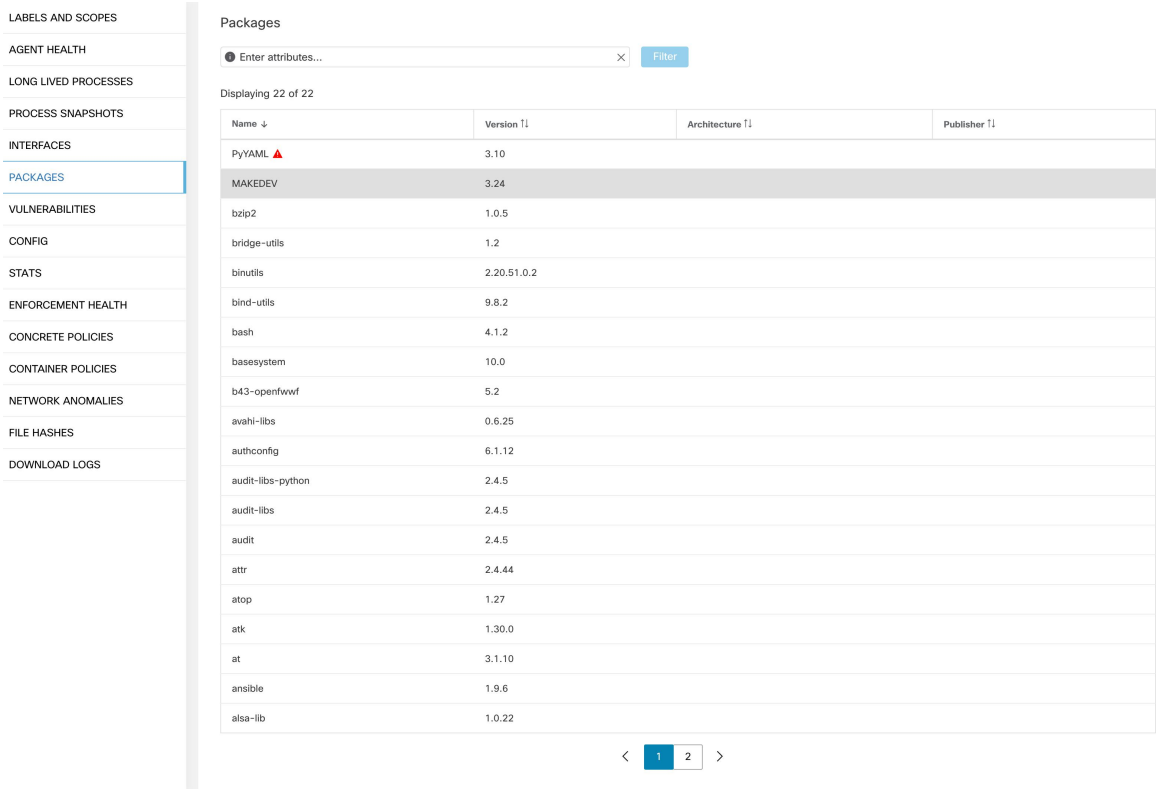
図 *222 :* ワークロード インターフェイスのリスト



[ソフトウェアパッケージ（Software Packages）] タブ

このタブには、ホストにインストールされているパッケージのリストが表示されます。ユーザーは、テーブルヘッダーのパッケージ属性に基づいて、ソフトウェアパッケージを選択して表示できます。

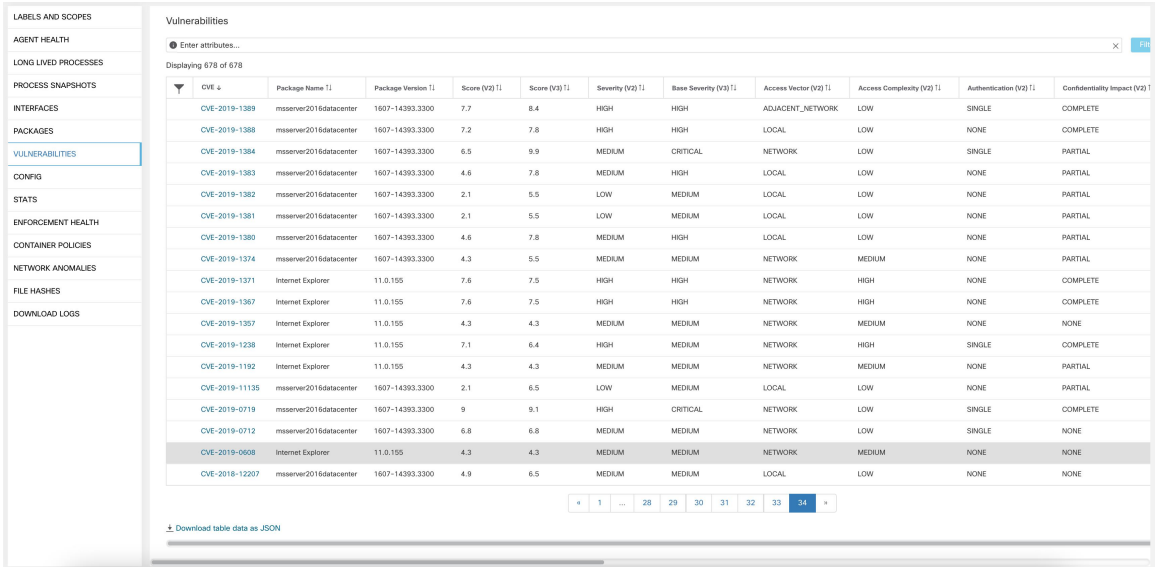
図 *223* : ソフトウェアパッケージ一覧



[脆弱性（Vulnerabilities）] タブ

このタブには、Common Vulnerabilities and Exposures（CVE）システムに基づいてワークロードで観察された検索可能な脆弱性が表示されます。「脆弱性データの可視化」を参照してください。

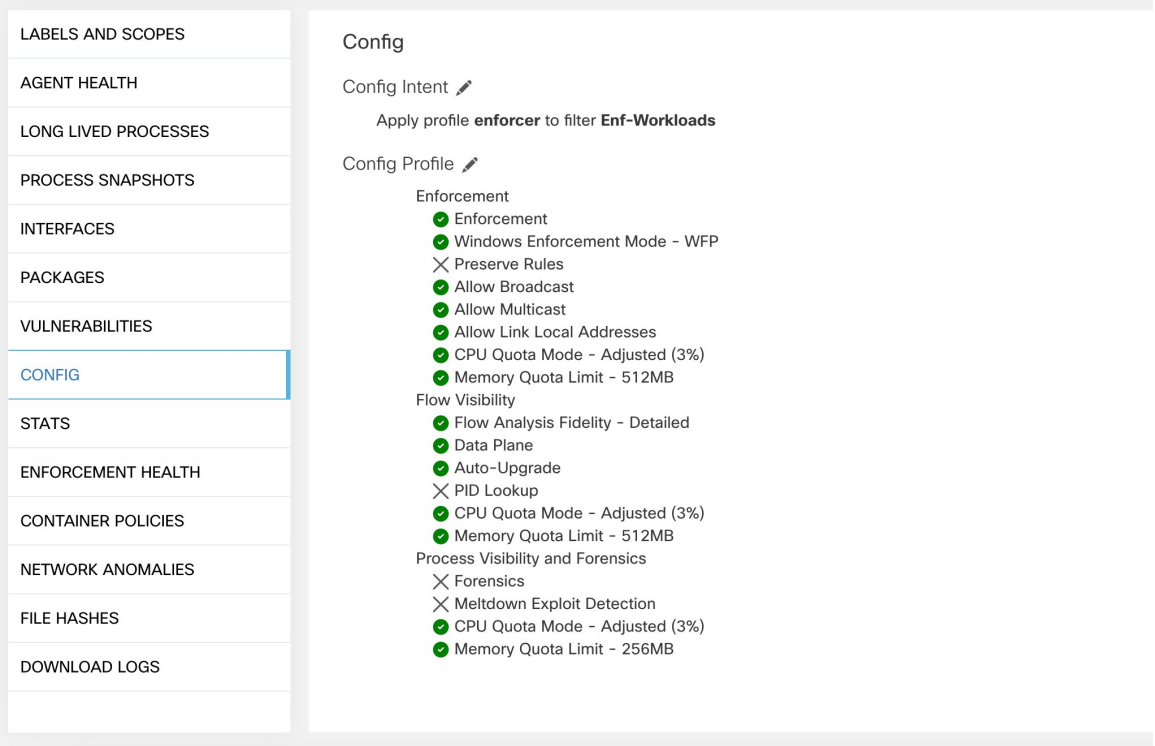
図 *224 : [*脆弱性（***Vulnerabilities***）*]* タブ



[エージェントの設定（Agent Configuration）] タブ

このタブには、ソフトウェアエージェントの設定が表示されます。優れた可視性エージェントおよび適用エージェントでのみ使用できます。これらの設定は、[エージェントの設定（Agent Configuration）]ページのエージェント設定インテントを使用して変更できます。「ソフトウェアエージェントの設定」を参照してください。

図 *225 :* 適用されるワークロード設定



[エージェント統計情報（Agent Statistics）] タブ

このタブには、ホストにインストールされている Secure Workload エージェントに関する統計情報が表示されます。優れた可視性エージェントおよび適用エージェントでのみ使用できます。

図 *226 :* エージェント統計情報



[具体的なポリシー（Concrete Policies）] タブ

ワークスペースが適用されると、各ワークロードは、そのワークロードに固有のそのワークスペース内のポリシーのみを受け取ります。各ワークロードで実際にプログラムされているこれらのポリシーは、具体的なポリシーと呼ばれます。

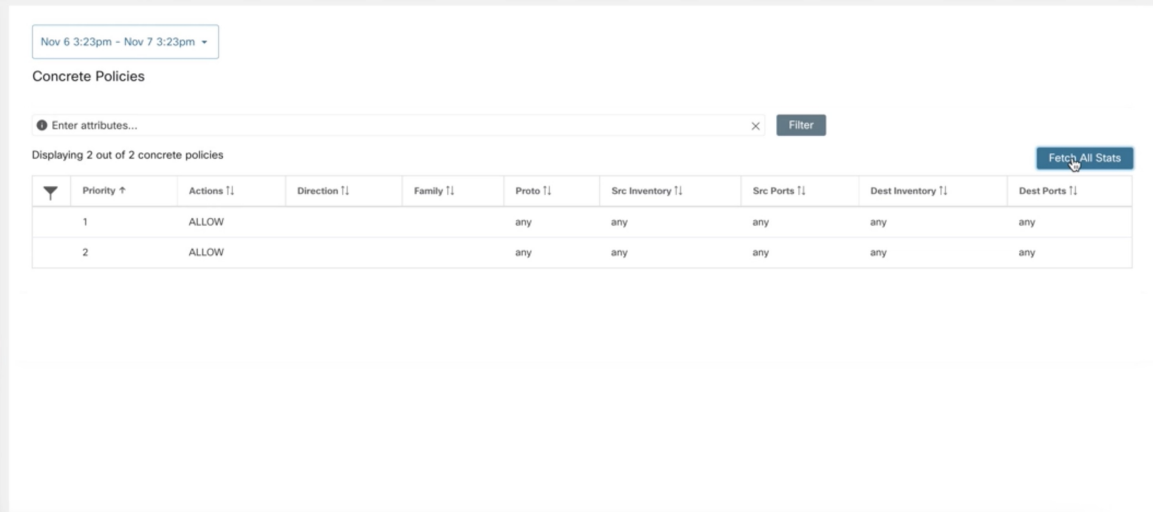
たとえば、アクションが ALLOW であるポリシーで指定されているプロバイダーに、サブネット 1.1.1.0/24 のすべてのインベントリが含まれているとします。このポリシーが Secure Workload エージェントを使用してワークロードにインストールされ、IP アドレス 1.1.1.2 が含まれている場合、ファイアウォールルールは次のようになります。

1. 着信トラフィックの場合、ファイアウォールルールは、サブネット 1.1.1.0/24 全体ではなく、厳密に 1.1.1.2 宛てのトラフィックのみを許可します。
2. 発信トラフィックの場合、ファイアウォールルールは、サブネット 1.1.1.0/24 全体からではなく、厳密に 1.1.1.2 からのトラフィックのみを許可します。

ワークロードプロファイルの [具体的なポリシー（CONCRETE POLICIES）] タブには、ホストに適用されている Secure Workload の具体的な適用ポリシーが表示されます。この表の各行は、ホストに実装されているファイアウォールルールに対応しています。各ポリシー行をさらに展開して、この具体的なポリシーが派生した元の論理インテントを表示することができます。ルールごとにパケット数とバイト数の時系列表示も可能です。[すべての統計情報の取得（Fetch

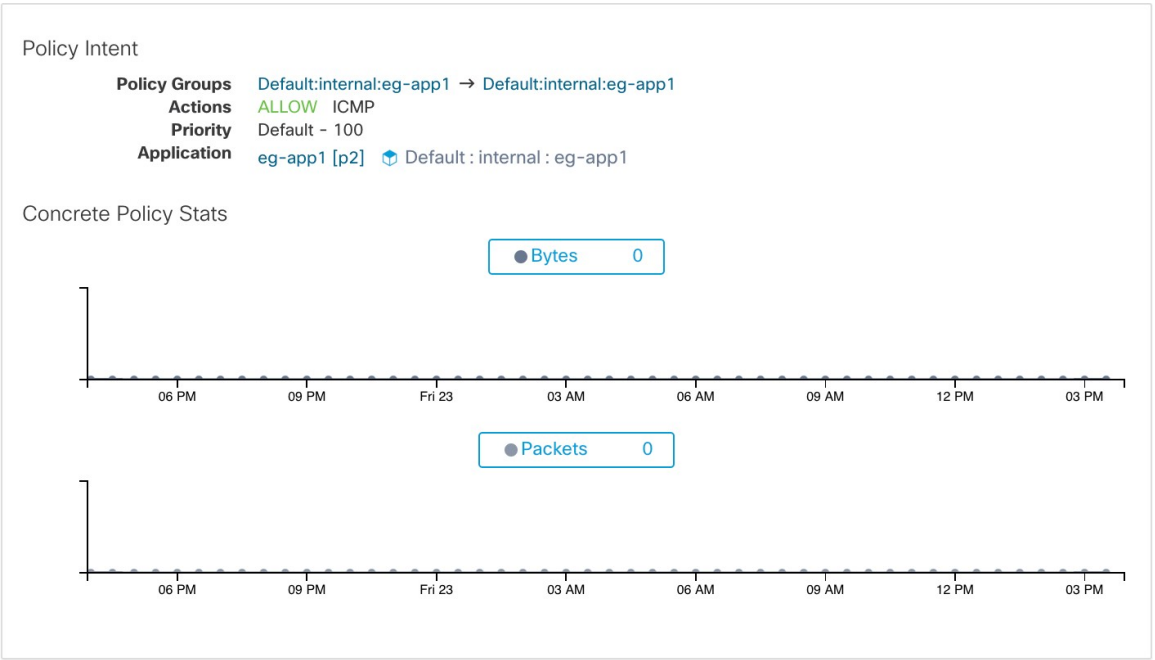
All Stats）] ボタンをクリックすると、各ルールのパケット数とバイト数が表示されます。このタブではフィルタを使用して、以下の表ヘッダーに示されているポリシーの属性に基づいて、適用されるポリシーのリストを絞り込むこともできます。このタブは、インストールされたエージェントで適用が有効になっている場合にのみ使用できます。

図 *227 :* 具体的なポリシーのリスト



次の画像では、[ポリシーグループ（Policy Groups）] にコンシューマとプロバイダーが表示されています。

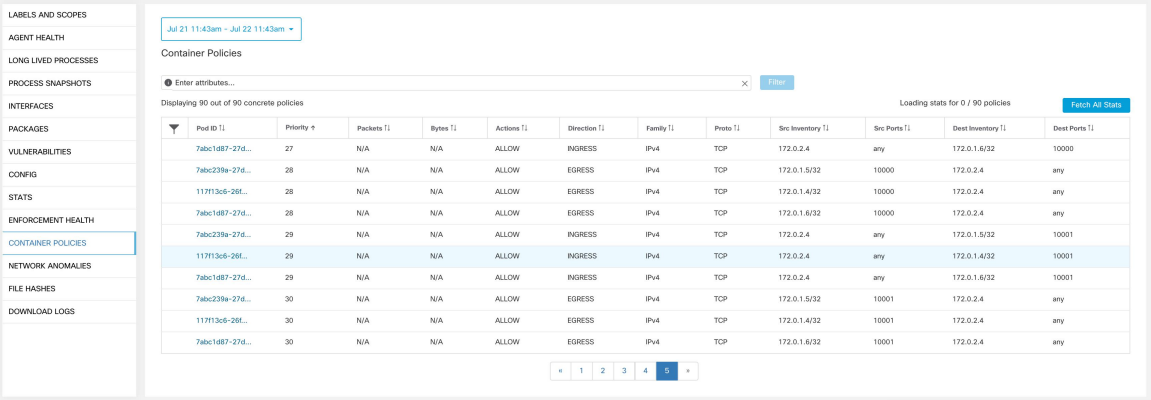
図 *228 :* 具体的なポリシーの行



[コンテナポリシー（Container Policies）] タブ

このタブには、コンテナに適用される Secure Workload の具体的な適応ポリシーが表示されます。この表の各行は、コンテナポッドに実装されているファイアウォールルールに対応しています。

図 *229* : コンテナの具体的なポリシー一覧



[ネットワーク異常（Network Anomalies）] タブ

このタブは、このワークロードに出入りする大規模なデータの移動を伴うイベントを識別するのに役立ちます。詳細については、「PCR ベースのネットワーク異常検出」を参照してください。

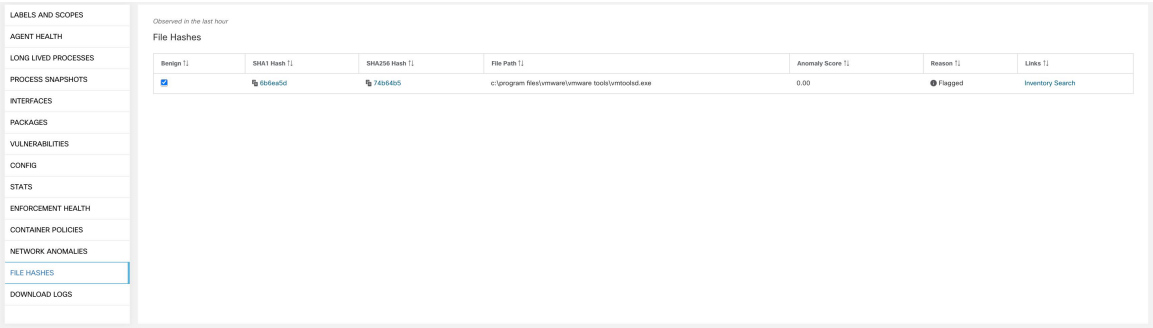
図 *230* : ワークロードネットワークの異常



[ファイルハッシュ（File Hashes）] タブ

このタブは、システム全体のプロセスバイナリハッシュの一貫性を評価することにより、プロセスハッシュの異常を検出します。詳細については、「プロセスハッシュの異常検出」を参照してください。

図 *231* : ワークロードファイルのハッシュ



Software Packages

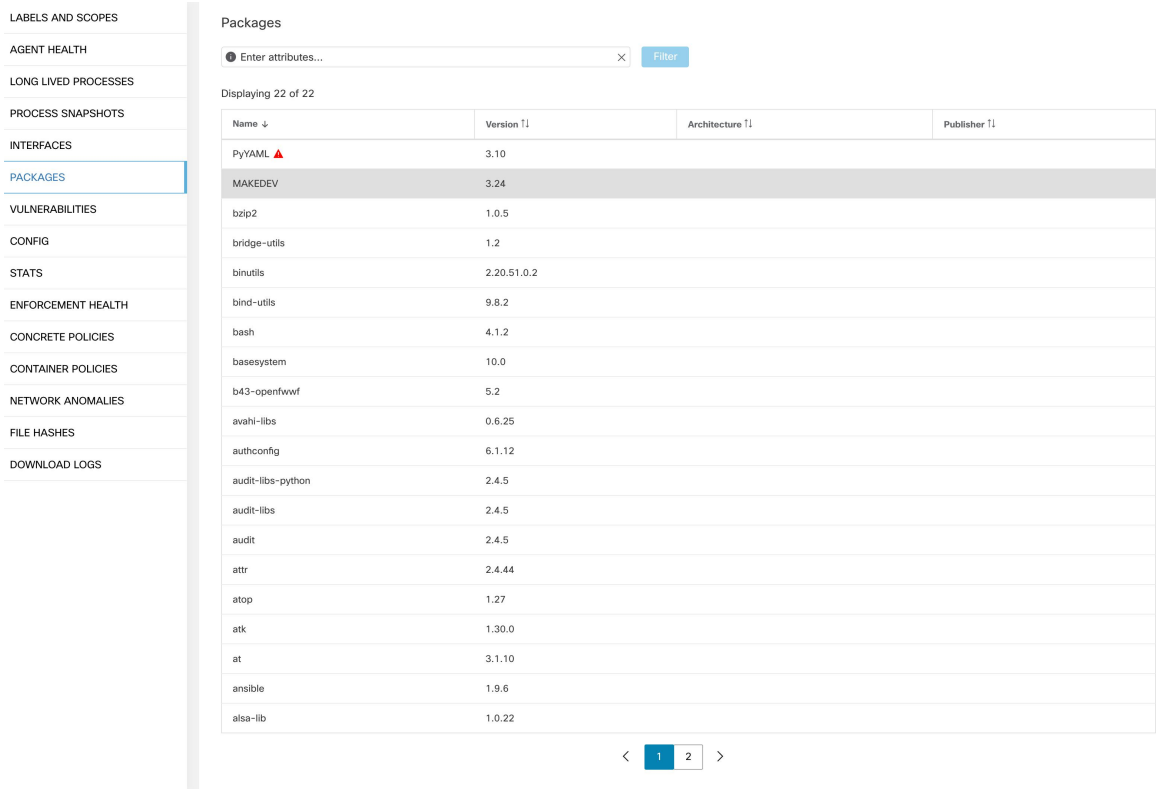
ソフトウェアパッケージ機能セットを使用すると、ホストにインストールされているパッケージと、それらに影響を与える脆弱性を表示できます。具体的には、次のことが可能になります。

- 次のパッケージマネージャに登録されているパッケージを表示します。
  - Linux：Redhat パッケージマネージャ（RPM）および Debian パッケージマネージャ（dpkg）
  - Windows：Windows レジストリサービス
- ホストにインストールされているパッケージに影響を与える Common Vulnerabilities and Exposures（CVE）を表示します。
- パッケージ名とバージョンを使用してインベントリフィルタを定義します。

[Packages] タブ

ホストにインストールされているパッケージを表示するには、ワークロード プロファイルの ワークロード プロファイル ページの [パッケージ（packages）] タブに移動します。

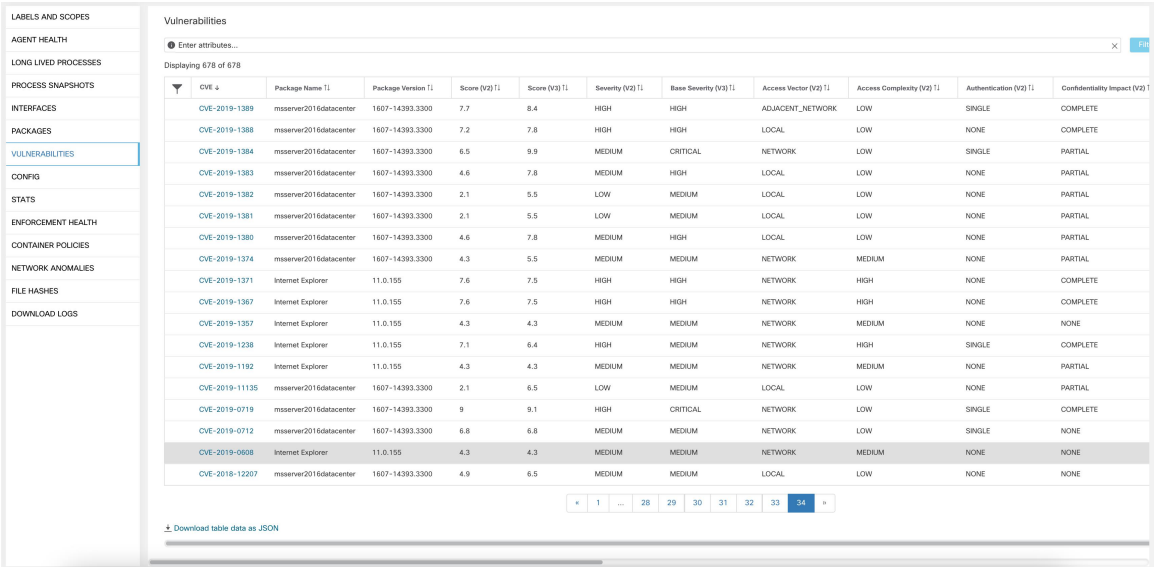
図 *232 :* ワークロード プロファイル パッケージ



Common Vulnerabilities and Exposures（CVE）

[パッケージ（Packages）] タブでは、パッケージが表示されるだけでなく、パッケージに影響を与える一般的な脆弱性とその重大度も表示されます。各脆弱性には、特定の脆弱性に関する詳細情報を提供する Nation Vulnerability Database（NVD）へのリンクが含まれています。CVE ID の表示に加えて、脆弱性の重大度を示す影響スコア（10 段階）も表示されます。

図 *233* : ワークロード プロファイル パッケージ *CVE*



Windows パッケージと CVE

次のセクションでは、Cisco Secure Workload へのパッケージ情報の報告に関する Windows エージェントの動作について説明します。

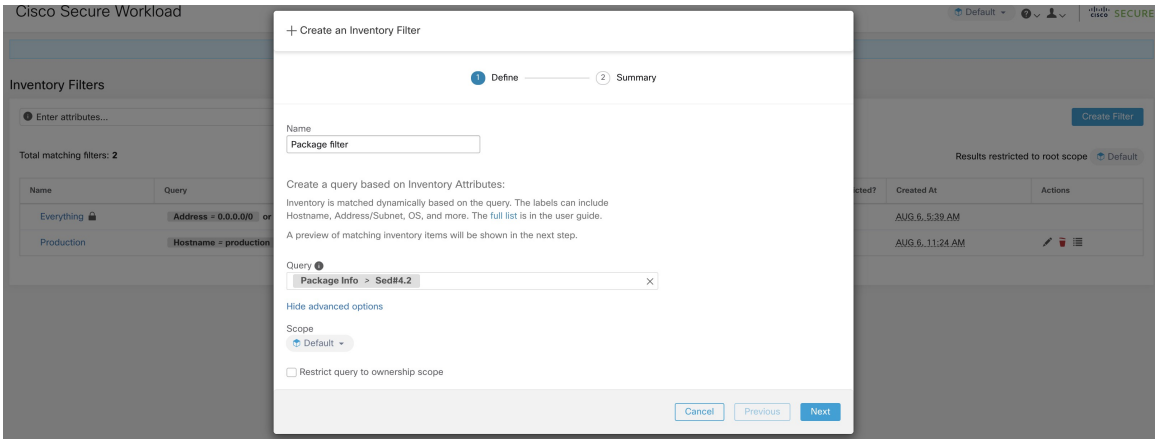
- Windows アプリケーション、PowerShell、IE はパッケージとして報告されます。.net フレームワークもパッケージとして報告されます。
- notepad.exe、cmd.exe、mstsc.exe など、その他の Windows アプリケーションは報告されません。
- Windows サーバーで設定されたロールと機能はパッケージとして報告されますが、バージョンが正しくない可能性があります。例：DNS サーバーが設定されている場合、報告されるバージョンは 0 または 8 のいずれかになります。
- Windows エージェントは、MSI インストーラまたは exe インストーラを使用してインストールされたサードパーティ製品を報告します。
  - MSI インストーラの場合、MSI API を使用してパッケージ情報（バージョン、発行元、パッケージ名など）を取得します。
  - exe インストーラを使用してパッケージをインストールする場合、パッケージ情報はレジストリから取得されます。
  - バージョン、発行元などの [パッケージインストーラ（Package installer）] フィールドはオプションです。バージョンが不明な場合、パッケージは報告されません。
  - 製品が zip ファイルから抽出されているか、アプリとしてインストールされている場合、パッケージリストで報告されません。

インベントリフィルタ

パッケージ名とバージョン（オプション）でインベントリフィルタを定義することで、パッケージ関連の情報を検索できます。

このフィルタのシンタックスは、PackageName#PackageVersion です。

図 *234* : インベントリパッケージ



次の操作がサポートされます。

- 等号：PackageName と PackageVersion（指定した場合）が一致するパッケージを搭載したホストを返します。
- 等号否定：PackageName は一致するが PackageVersion（指定した場合）は一致しないパッケージを搭載したホストを返します。
- 超過：PackageName が一致し、バージョンが PackageVersion より大きいパッケージを搭載したホストを返します。
- 以上：PackageName が一致し、バージョンが PackageVersion 以上のパッケージを搭載したホストを返します。
- 未満：PackageName が一致し、バージョンが PackageVersion 未満のパッケージを搭載したホストを返します。
- 以下：PackageName が一致し、バージョンが PackageVersion 以下のパッケージを搭載したホストを返します。

脆弱性データの可視化

脆弱性データの可視化機能により、ホスト上のパッケージとプロセスに影響を与える脆弱性を検出して表示できます。インベントリフィルタは、次を使用して定義できます。

```
- CVE IDs.- CVSS v2 and v3 scores.- CVSS v2 access vector and access complexity.- CVSS
v3 attack vector, attack complexity, and privilege required.
```

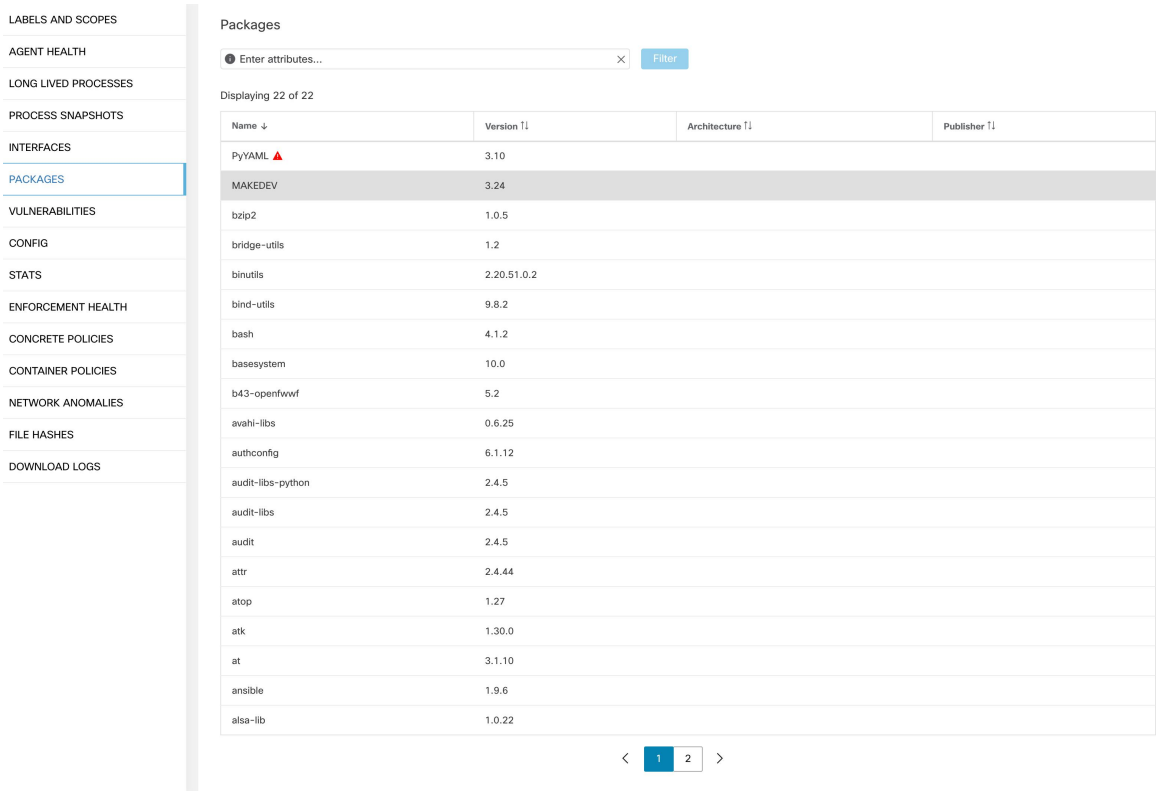
ワークロード プロファイル ページ

システム上のパッケージとプロセスに影響を与える脆弱性関連の情報は、ワークロード プロファイル ページに表示されます。

[Packages] タブ

[パッケージ（Packages）] タブには、ホストにインストールされているパッケージと、それらに影響を与える脆弱性が一覧表示されます。

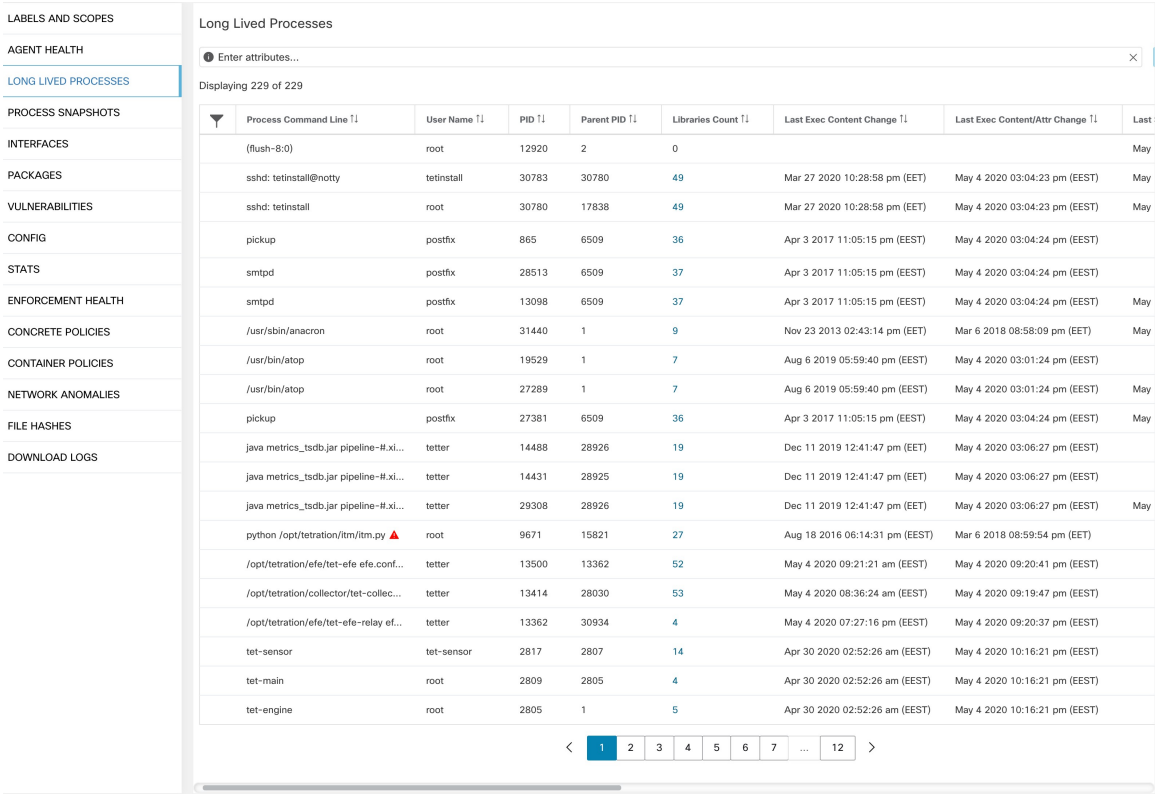
図 *235 :* ワークロード プロファイル パッケージ



[プロセスリスト（Process List）] タブ

存続期間の長いプロセスは、プロセスリストタブに表示されます。

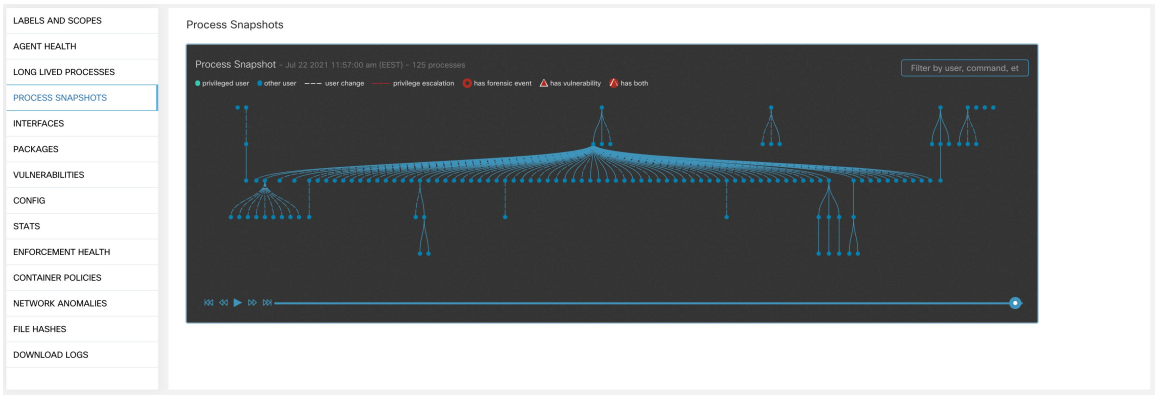
図 *236 :* ワークロードプロファイルのプロセスリスト



[プロセススナップショット（Process Snapshot）] タブ

[プロセススナップショット（Process Snapshot）] タブの下の下にあるプロセスツリーには、すべてのプロセスに関する脆弱性情報が表示されます。

図 *237 :* ワークロードプロファイルの *[プロセススナップショット（Process Snapshot）]* タブ



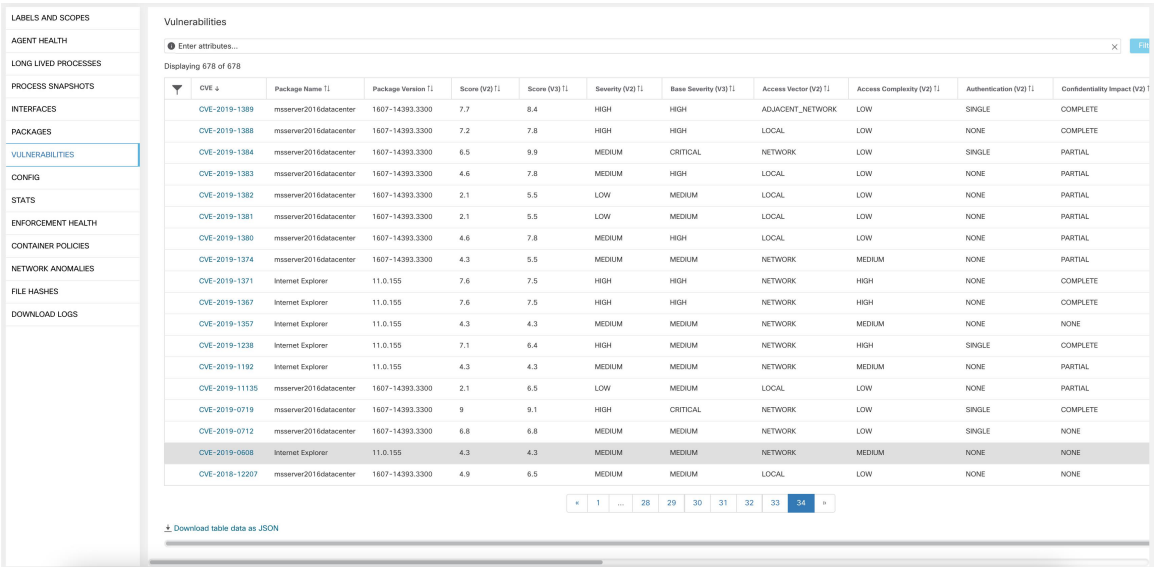
[脆弱性（Vulnerabilities）] タブ

[脆弱性（vulnerability）] タブには、ワークロードで観察された脆弱性のリストが表示されます。

CVE ごとに、基本的な影響指標に加えて、脅威インテリジェンスに基づくエクスプロイト情報が表示されます。

- エクスプロイト数：昨年、CVE が実際に悪用されたのが確認された回数
- 最終エクスプロイト：脅威インテリジェンスによって、CVE が実際に悪用されたのが最後に確認された時間

図 *238 :* ワークロードプロファイルの脆弱性タブ



インベントリフィルタ

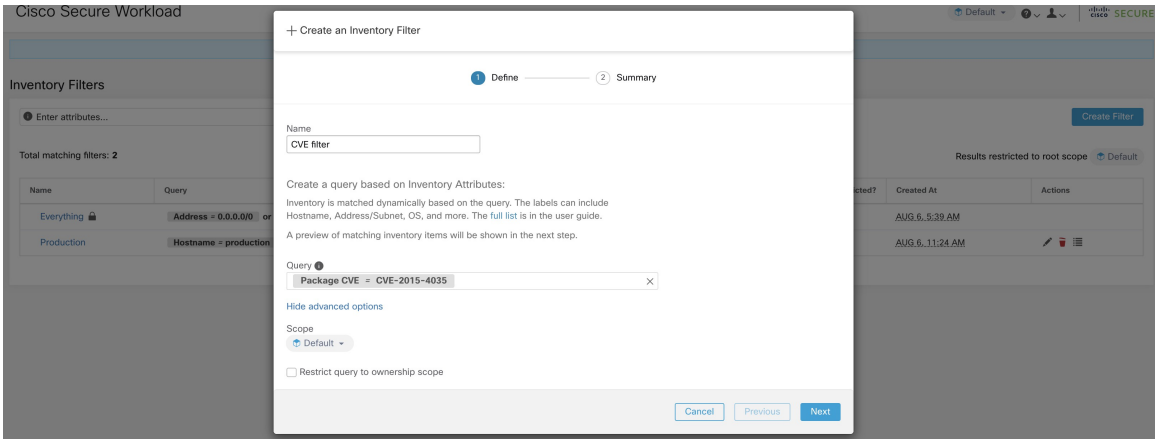
次のタイプのインベントリフィルタを定義して、脆弱なパッケージを持つホストを特定できます。

CVE ID ベースのフィルタ

このフィルタにより、特定の CVE または任意の CVE の影響を受けるホストを検索できます。

特定の CVE の影響を受けるホストを検索するには、CVE ID を CVE-XXXX-XXXX の形式で指定します。

図 *239* : インベントリフィルタ *CVE*



次の操作がサポートされます。

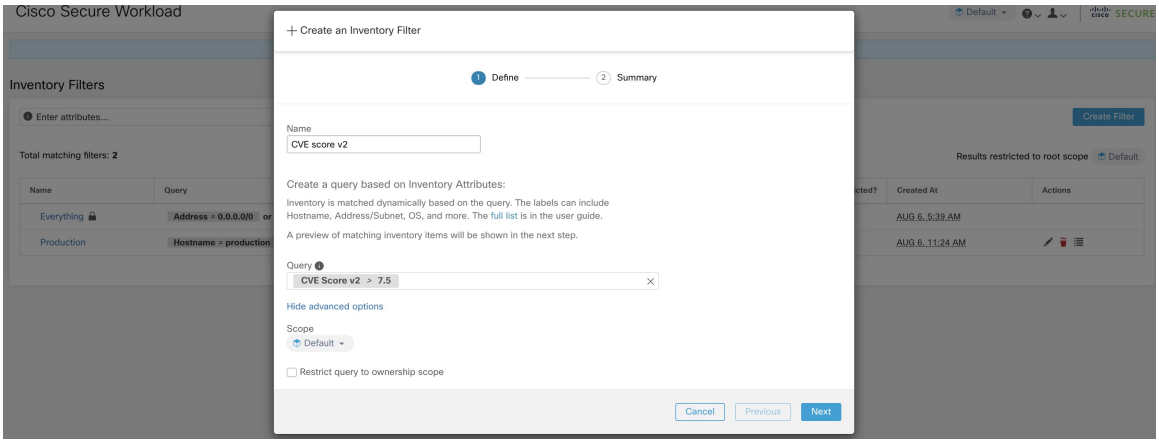
- [等価（Equality）]：CVE ID の影響を受けるパッケージを持つホストを返します。
- [非等価（Inequality）]：CVE ID の影響を受けないパッケージを持つホストを返します。
- [含む（Contains）]：入力文字列に存在する CVE の影響を受けるパッケージを持つホストを返します（「cve」と入力すると、CVE の影響を受けるホストが返されます）。
- [含まない（Doesn't contain）]：入力文字列に存在する CVE の影響を受けないパッケージを持つホストを返します（「cve」と入力すると、CVE の影響を受けないホストが返されます）。

CVSS（共通脆弱性評価システム）インパクトスコアベースのフィルタ

このフィルタにより、指定された CVSSv2 または CVSSv3 影響スコアがある CVE を持つホストを検索できます。影響スコア（v2 または v3）の CVE を持つホストを検索するには、スコアを数値形式で指定します。

CVSSv2 影響スコアが 7.5 を超える CVE を持つホストを検索します。

図 *240* : インベントリフィルタ *CVSS*



次の操作がサポートされます。

- [等価（Equality）]：指定された CVSSv2 または CVSSv3 影響スコアがある CVE を持つホストを返します。
- [非等価（Inequality）]：指定された CVSSv2 または CVSSv3 影響スコアがある CVE を持たないホストを返します。
- [より大きい（Greater Than）]：指定された CVSSv2 または CVSSv3 影響スコアそれぞれよりも大きい CVSSv2 または CVSSv3 影響スコアがある CVE を持つホストを返します。
- [以上（Greater Than or Equal To）]：CVSSv2 または CVSSv3 影響スコアがそれぞれ指定された CVSSv2 または CVSSv3 影響スコア以上である CVE を持つホストを返します。
- [より少ない（Less Than）]：指定された CVSSv2 または CVSSv3 影響スコアそれぞれよりも小さい CVSSv2 または CVSSv3 影響スコアがある CVE を持つホストを返します。
- [以下（Less Than or Equal To）]：CVSSv2 または CVSSv3 影響スコアがそれぞれ指定された CVSSv2 または CVSSv3 影響スコア以下である CVE を持つホストを返します。

CVSSv2 ベースのフィルタ

インベントリフィルタは、脆弱なホストを特定するため、Access Vector とアクセスの複雑さを使用して作成できます。これらのフィルタは、次のタイプの操作をサポートします。

- Equality：フィルタに一致する脆弱性の影響を受けるパッケージを持つホストを返します。
- Inequality：フィルターに一致する脆弱性の影響を受けないパッケージを持つホストを返します。

Access Vector

Access Vector は、脆弱性がどのようにエクスプロイトされるかを反映します。攻撃者が脆弱なシステムから遠ざかるほど、ベーススコアは高くなります。以下の表は、さまざまな Access Vector とそのアクセス要件を示しています。

値	アクセスの種類
LOCAL	物理またはローカル（シェル）。
ADJACENT_NETWORK	ブロードキャストまたはコリジョン。
NETWORK	リモートでエクスプロイト可能。

アクセスの複雑さ

このメトリックは、攻撃者が標的のシステムにアクセスできるようになった後に、脆弱性をエクスプロイトする際の複雑さを測定します。ベーススコアは、アクセスの複雑さに反比例します。さまざまなタイプのアクセスの複雑さは次のとおりです。

値	説明
HIGH	特殊なアクセス条件が存在します。
[中（Medium）]	アクセス条件はやや特殊です。
LOW	特殊なアクセス条件は存在しません。

CVSSv3 ベースのフィルタ

攻撃元区分、攻撃の複雑さ、および特権は、CVSSv3スコアに影響を与えるために必要であり、インベントリフィルタで使用できます。これらのフィルタは、次の操作をサポートします。

- Equality：フィルタに一致する脆弱性の影響を受けるパッケージを持つホストを返します。
- Inequality：フィルターに一致する脆弱性の影響を受けないパッケージを持つホストを返します。

攻撃元区分

このメトリックは、脆弱性のエクスプロイトが可能になるコンテキストを表します。攻撃者が脆弱なコンポーネントから遠ざかるほど、ベーススコアは高くなります。以下の表に、さまざまな攻撃元区分とそのアクセス要件を示します。

値	アクセスの種類
LOCAL	ローカル（キーボード、コンソール）またはリモート（SSH）。
PHYSICAL	物理的なアクセスが必要です。
ADJACENT_NETWORK	ブロードキャストまたはコリジョン。
NETWORK	リモートでエクスプロイト可能。

Attack Complexity

このメトリックは、脆弱性をエクスプロイトするために存在する必要がある条件を示します。ベーススコアは、最も複雑でない攻撃に対して最大です。さまざまなタイプのアクセスの複雑さは次のとおりです。

値	説明
HIGH	攻撃の設定と実行には多大な労力が必要です。
LOW	特殊なアクセス条件は存在しません。

必要な権限

このメトリックは、脆弱性のエクスプロイトを成功させるために攻撃者が持っている必要がある権限レベルを表しています。ベーススコアは、攻撃を実行するための権限が必要ない場合に最大です。必要な権限のさまざまな値は次のとおりです。

値	必要な権限
HIGH	脆弱なコンポーネントに対する重要な制御を提供する権限。
LOW	機密でないリソースへのアクセスを許可する低い権限。
NONE	攻撃の実行に特権は必要ありません。

サービス プロファイル

Cisco Secure Workload は、外部オーケストレータを介して取り込まれたすべての Kubernetes サービスや他のロードバランサの可視性を提供します。サービスプロファイルページには、特定のサービスの詳細が表示されます。



（注） サービスプロファイルページはさまざまな場所からリンクされています。サービスプロファイルを表示する方法の 1 つは、検索で説明されているようにサービスの検索を実行することです。

検索結果から、[サービス（Services）] タブの下の [サービス名（Service Name）] をクリックして、そのプロファイルに移動します。このサービスについては、次の情報が利用できます。

ヘッダー（Header）

ヘッダーは次の情報で構成されます。

- [オーケストレータ名（Orchestrator Name）]：このサービスを報告した外部オーケストレータの名前。
- [オーケストレータタイプ（Orchestrator Type）]：外部オーケストレータのタイプ。
- [名前空間（Namespace）]：サービスの名前空間。
- [サービスタイプ（Service Type）]：サービスのタイプ。値は、ClusterIP、NodePort、LoadBalancer のいずれかです。

IP とポート

この表には、このサービスにアクセスできる IP とポートのすべての可能な組み合わせが記載されています。NodePort タイプのサービスの場合、この表には ClusterIP:Port と NodeIp:NodePort の両方の関連付けが表示されます。

ユーザーラベル

このサービスに対してユーザーがアップロードしたラベルおよびオーケストレータ システムによって生成されたラベルのリスト。

範囲

ポッドが属する範囲のリスト。

ポッドプロファイル

Cisco Secure Workload は、Kubernetes 外部オーケストレータを介して取り込まれたすべての Kubernetes ポッドの可視性を提供します。ポッドプロファイルページには、特定のポッドの詳細が表示されます。

（注） ポッドプロファイルページはさまざまな場所からリンクされています。ポッドプロファイルを表示する方法の 1 つは、検索で説明しているようにポッドの検索を実行することです。

検索結果から、[ポッド（Pod）] タブの下の [ポッド名（Pod Name）] をクリックして、そのプロファイルに移動します。ポッドに関する次の情報を入手できます。

ヘッダー（Header）

ヘッダーは次の情報で構成されます。

- [オーケストレータ名（Orchestrator Name）]：このポッドを報告した外部オーケストレータの名前。
- [オーケストレータタイプ（Orchestrator Type）]：外部オーケストレータのタイプ。
- [名前空間（Namespace）]：ポッドの名前空間。

• [IPアドレス（IP Address）]：ポッドの IP アドレス。

ユーザーラベル（User Labels）

このポッドに対してユーザーがアップロードしたラベルおよびオーケストレータ システムによって生成されたラベルのリスト。

スコープ

サービスが属する範囲のリスト。

近隣



（注） 近隣は廃止され、次の Cisco Secure Workload リリースで削除されます。

近隣アプリケーションを使用すると、ユーザーは集約されたフローデータを地理的位置ごとに、またはノード間のエッジやパスなどのノード周辺の近隣について調べることができます。近隣アプリケーションでは、地理関連のアラートや、ノード、エッジ、ホップベースのアラートなど、いくつかのタイプのアラートを設定することもできます。



（注） **前提条件**：

1. 範囲階層を作成するか、ポリシーを自動的に検出して、ライブ分析を有効にして実際に近隣グラフを表示します。グラフを表示するには、近隣はフロー上でサブ範囲、フィルタ**、またはクラスタ** に注釈が付けられている必要があります。** フィルタとクラスタは、ライブ分析または適用が有効になっているプライマリワークスペースの一部である必要があり、そのワークスペースの範囲の一部である必要があります。
2. 近隣地理情報には、脅威インテリジェンスを介して地域データパックがアップロードされている必要があります。
3. 近隣地理情報では、ユーザーの Web ブラウザは、マップのレンダリングのために Mapbox API へのアクセス権が必要です。

アクセス

近隣には、左側のメニューの [調査（investigate）] からアクセスできます。

有効化/無効化の方法

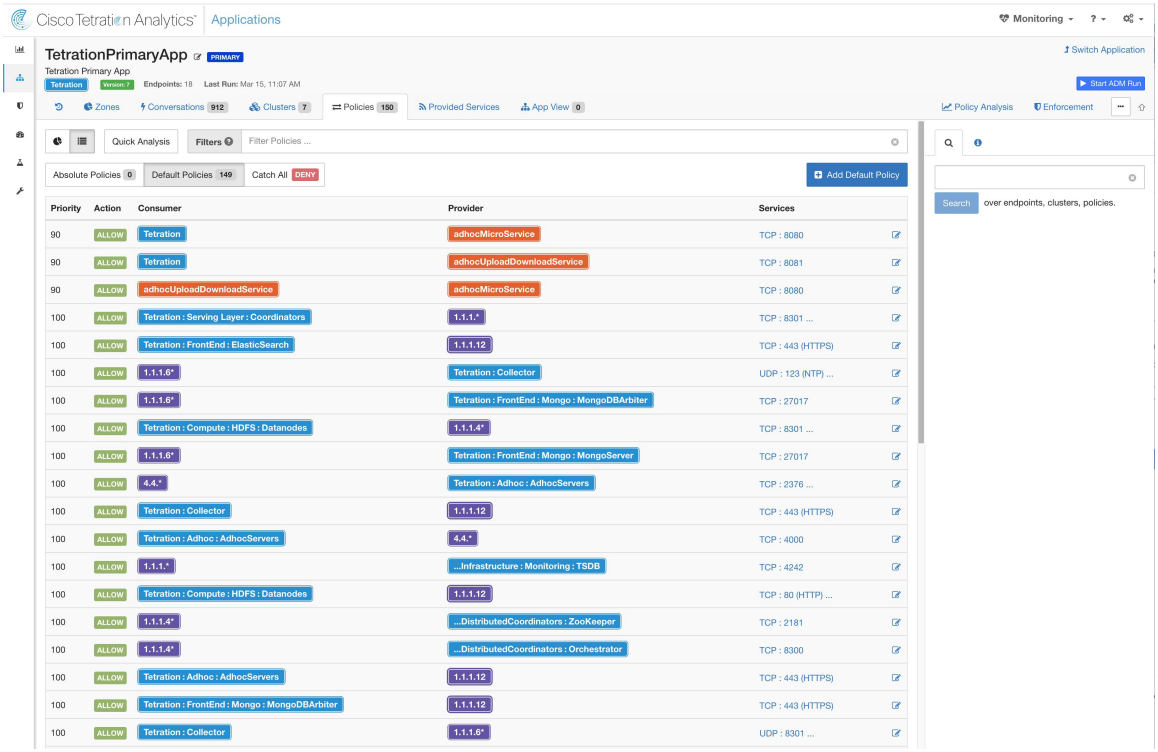
近隣は、すべてのルート範囲で自動的に有効になります。

用語

ノード

- ノードは、ライブ分析または適用が有効になっているプライマリ アプリケーション ワークスペースの一部である範囲またはインベントリフィルタ/クラスタです。さらに、フィルタには、ライブ分析または適用が有効になっているワークスペースに対応する所有範囲が必要です。

図 *241* : アプリケーションライブ分析



制限事項

- インベントリフィルタとクラスタについて、個々の範囲には 500 のサイズ制限があります。
- 優先順位は、最新のインベントリフィルタ、次に最新のクラスタの順に与えられます。最新は、更新時間で決まります。

近隣データの探索

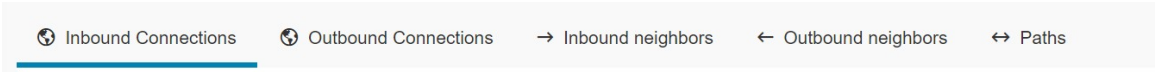
「Neighborhood」アプリをクリックすると、ビューが Neighborhood UI に変わり、近隣データを探索することができます。

Neighborhood には 5 つの異なるビューがあります。

1. 地理インバウンド
2. 地理アウトバウンド
3. インバウンドネイバー

4. アウトバウンドネイバー
5. パス

図 *242 :* 近隣探索オプション



地理データの探索

近隣地域地理情報は、地理データを 2 つの方向で提供します。

1. インバウンド：特定の地理的位置（国など）から特定の範囲（つまりフィルタ/クラスタ）へのフローの集約ビュー。
2. アウトバウンド：特定の範囲（つまりフィルタ/クラスタ）から特定の地理的位置（国など）へのフローの集約ビュー。

地理ビューは、データの方向ではなく、送信元/コンシューマ範囲（アウトバウンド）または宛先/プロバイダー範囲（インバウンド）に基づいていることに注意してください。選択した地理ビュー内で、[送信バイト数（Bytes Sent）] または [受信バイト数（Bytes Received）] を選択できます。

図 *243 :* インバウンド：地理的位置 → ノード

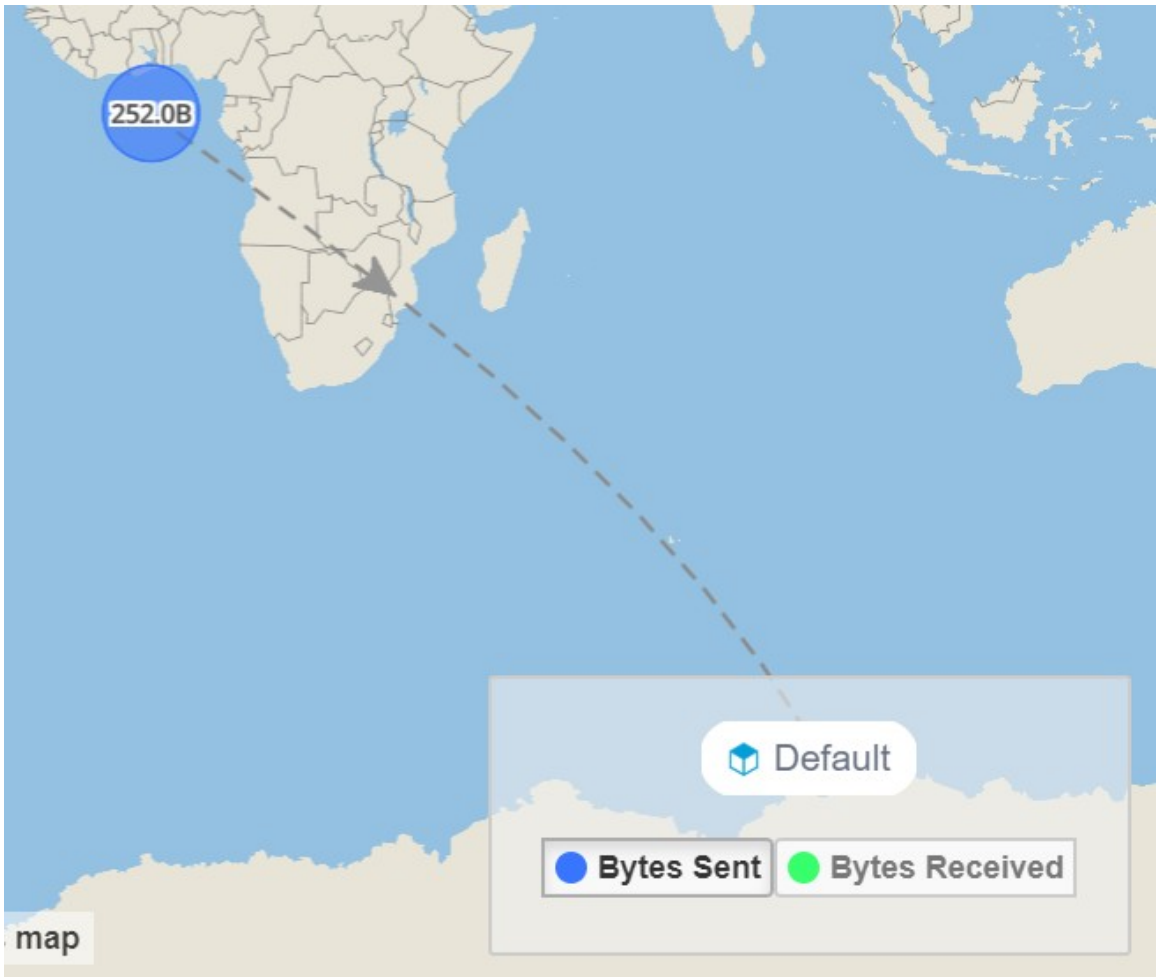
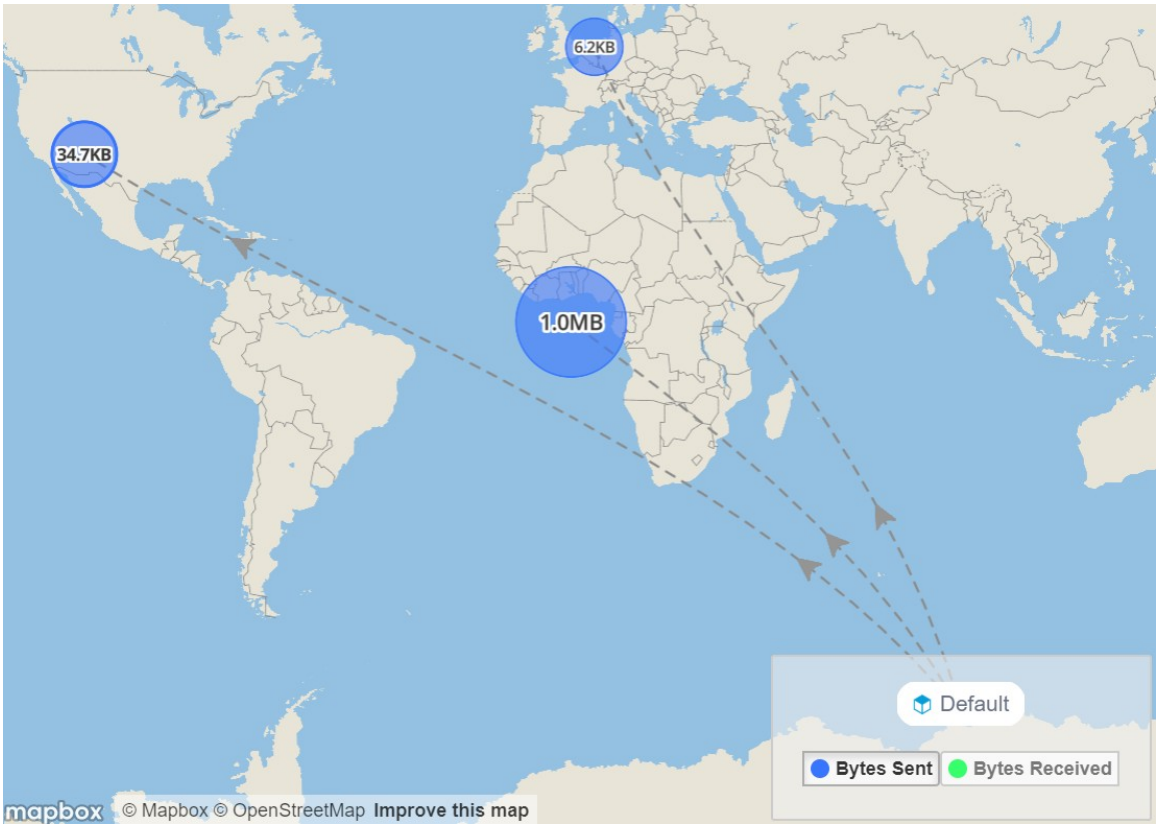
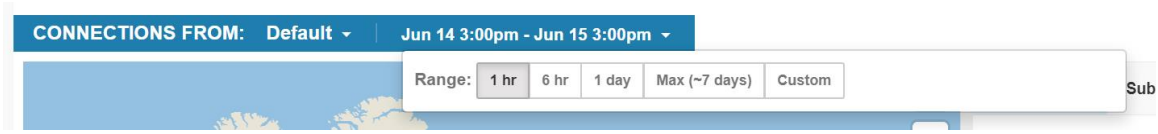


図 *244 :* アウトバウンド：ノード → 地理的位置



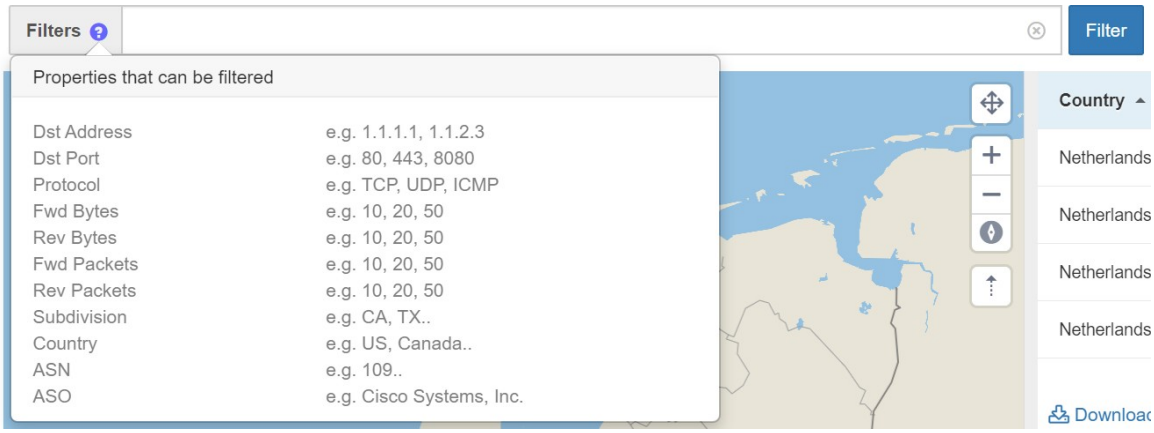
サポートされているフィルタ

図 *245 :* データを集計する時間範囲を選択するためのオプション



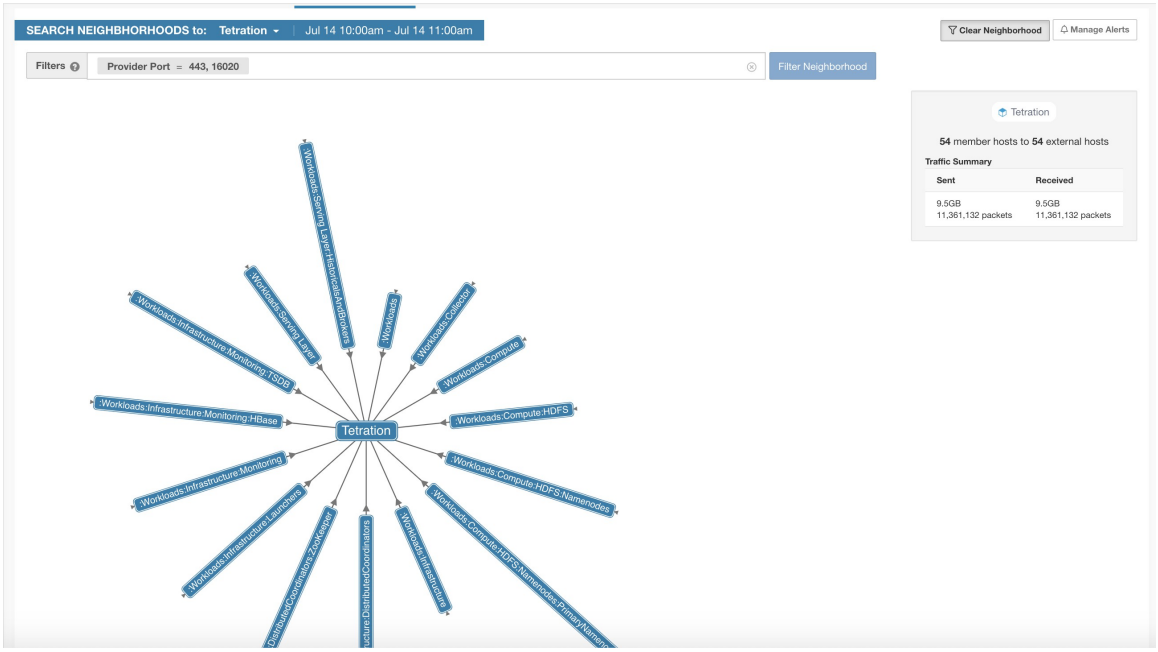
（注） 過去 7 日間まで。

図 *246 :* 近隣データをフィルタリングするためのオプション



フィルタ入力機能は、「-」を範囲クエリに変換することで、ポート、コンシューマアドレス、プロバイダーアドレスの「,」と「-」もサポートします。以下は、有効なフィルタの例です。

図 *247 :* 例：フィルタ入力機能はポートの「,」をサポートします



ナビゲーション

[地理（Geo）] ページをナビゲートするための主なポイント：

手順

ステップ 1 [インベントリプロファイル（Inventory Profile）] に移動して、より多くの過去の地理データを含む詳細なインベントリ情報を確認します。

ステップ 2 ノードを選択します。注：地理データが利用可能な範囲のみがドロップダウンリストに表示されます。

ステップ 3 時間範囲を選択します。

ステップ 4 フィルタ選択をオンまたはクリアに切り替えます。

ステップ 5 地理位置情報が不明なデータは、「ヌル島」からのデータ、または「ヌル島」に向かうデータとして表示されます。

ステップ 6 矢印にフローの送信元が範囲またはノード（ここに表示）であるか、または世界であるかが示されます。

ステップ 7 複数の地理位置情報がマップ上でグループ化される場合があります。手のひらアイコンは、クラスタをクリックして拡大し、曖昧さを解消できるかどうかを示します。

ステップ 8 表の行をクリックすると、国と区画がフィルタとして設定され、マップが拡大されて複数の住所が表示されます。

ステップ 9 マップをフルスクリーンモードにします。

ステップ 10 マウスの下にある地域を中心にマップを拡大します。

ステップ 11 マップを縮小します。

ステップ 12 ボタンを所定の位置にドラッグして、マップの方位を変更して、より 3D のような外観にします。

ステップ 13 表示されたポップアップとともに線と矢印をオフにして、データクラスタを強調します。

ステップ 14 マップの水平方向の**サイズを変更**して、マップまたはテーブルの表示を強調します。

その他の注意点：

- マップの右下に、選択した [ノード/範囲（Node/Scope）] が表示されます。[送信バイト数（Bytes Sent）] または [受信バイト数（Bytes Received）] を選択できます。
- 表の下部に、JSON データのダウンロードリンクが表示されます。

その他の注意点：

図 *248 :* 地理の強調表示されたナビゲーションポイント

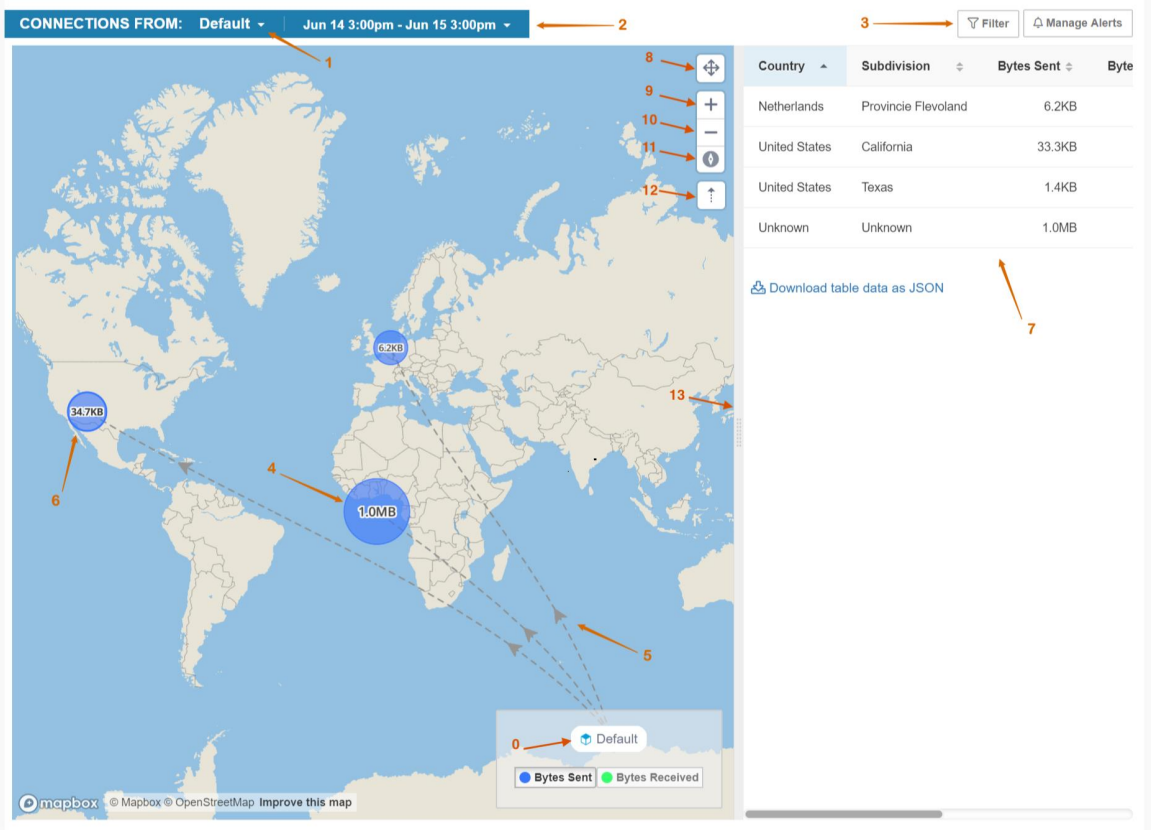
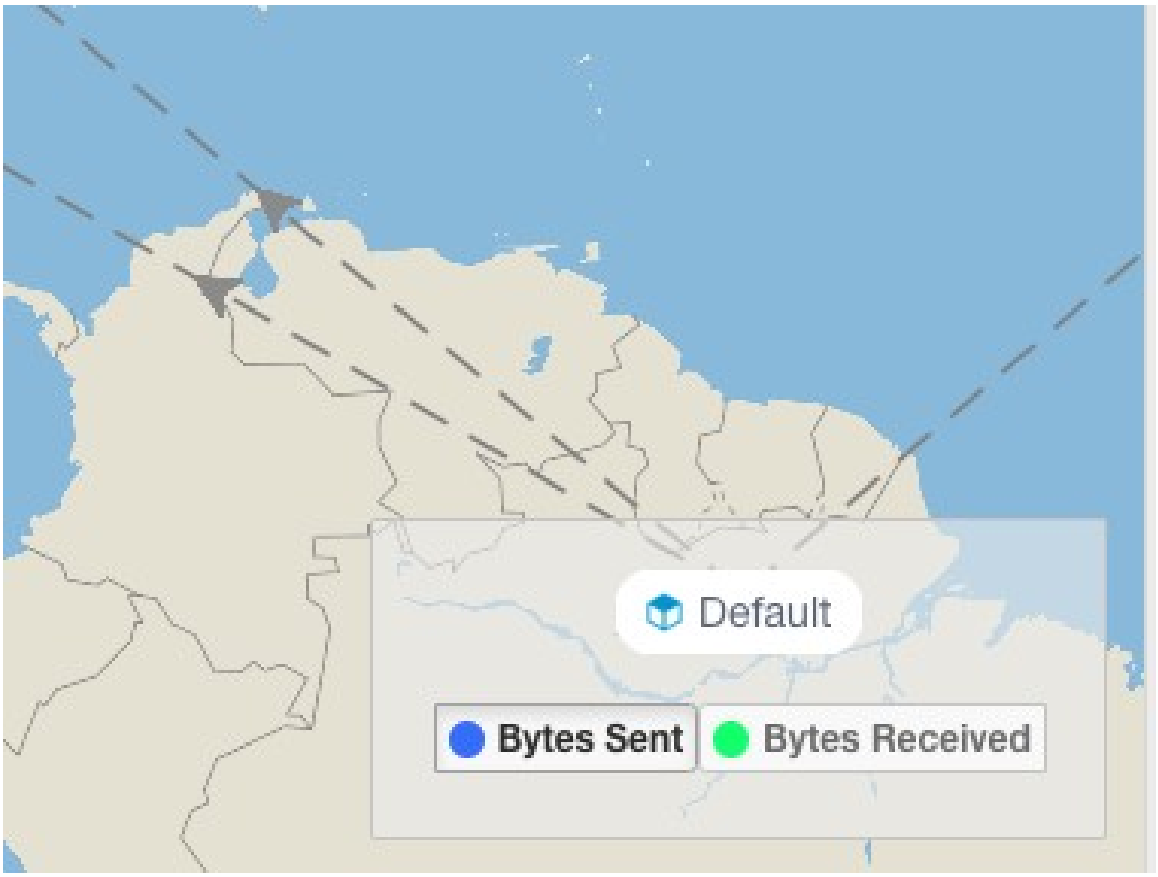
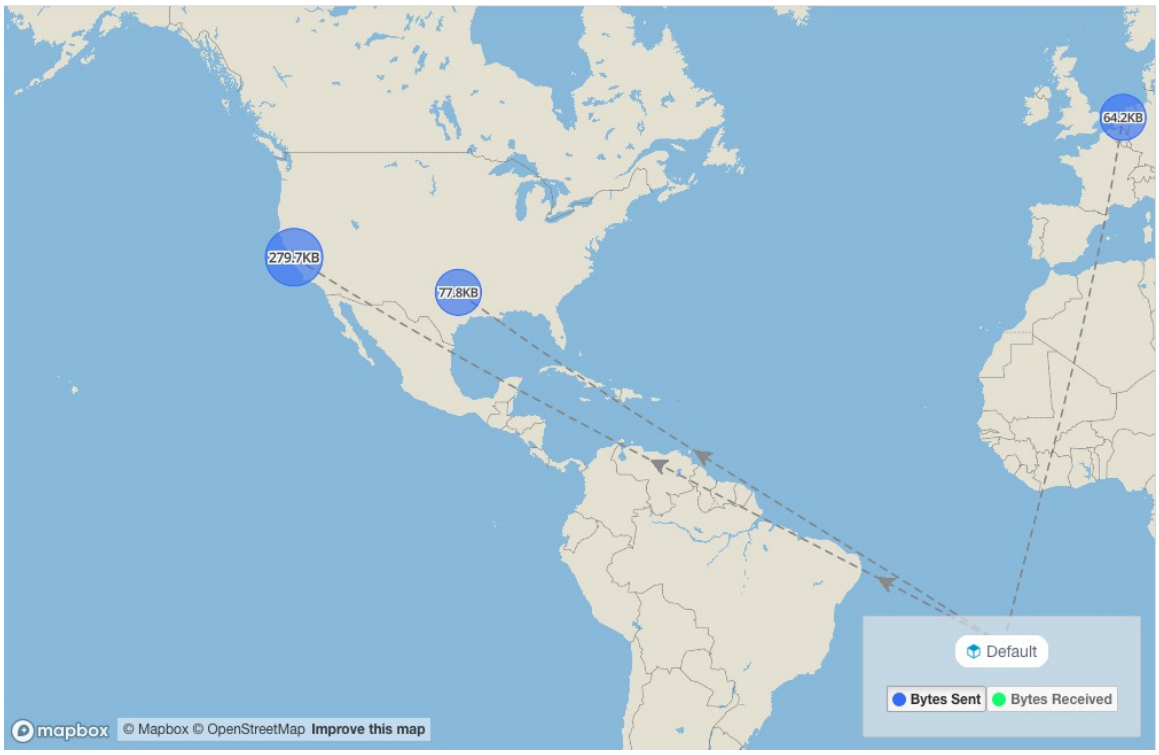


図 *249 : 例 0*



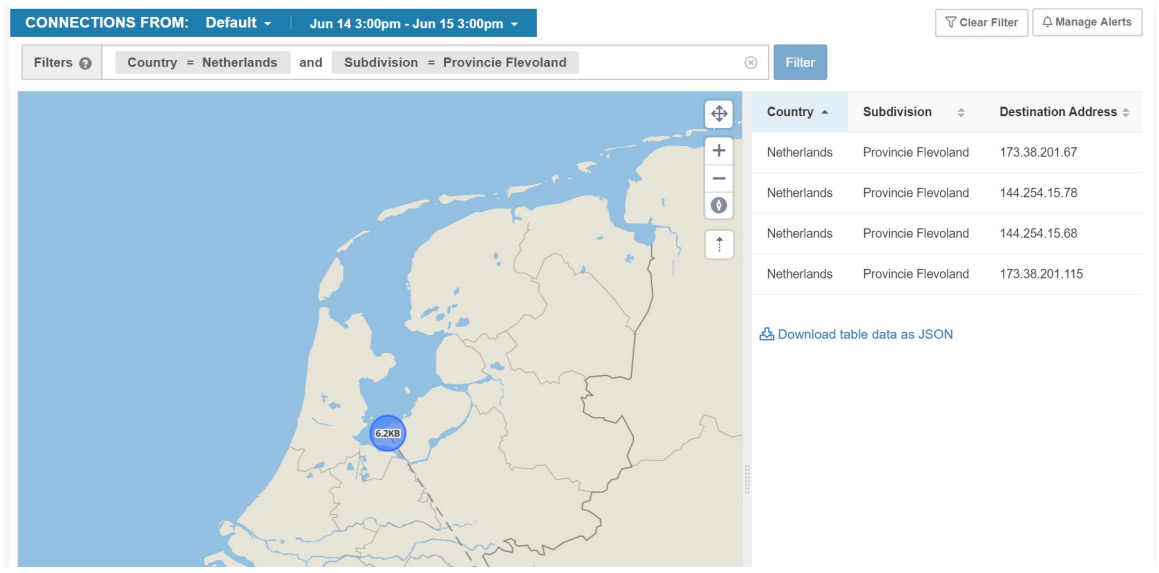
[ノード/範囲（Node/Scope）] をクリックすると、[インベントリプロファイル（Inventory Profile）] に移動します。インベントリプロファイル （460 ページ）

図 *250 : 例 6*



マップ上のクラスタ化されたポイントのグループ（手のアイコンで識別）をクリックすると、拡大されて複数のクラスタ化されたポイントが明確になります。

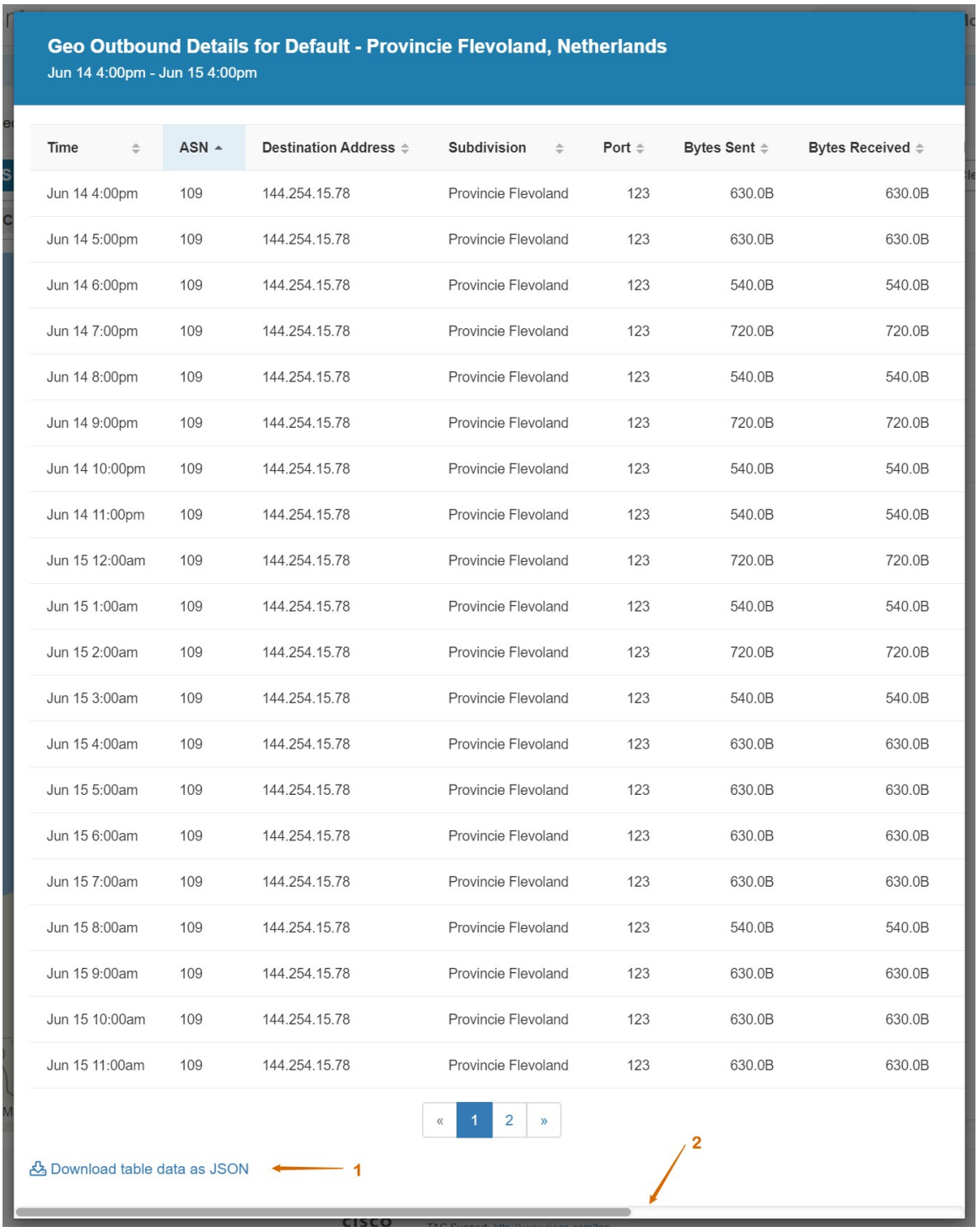
図 *251 : 例 7*



テーブルの行をクリックすると、プロパティがフィルタに設定され、拡大されて複数のアドレスがテーブルに表示されます。

特定の送信元と宛先を選択すると、詳細ビューがポップアップ表示されます。

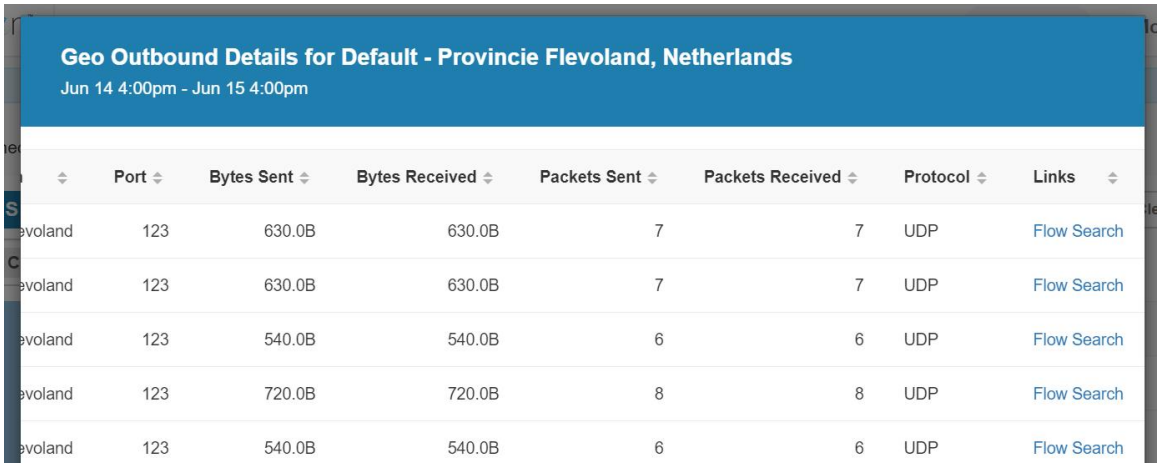
図 *252*：以前のアドレスリストビューの行をクリックして、詳細ビューをポップアップ表示する



1. このデータはダウンロードできます。

2. 右にスクロールして、フロー検索リンクなどの追加の列に移動します。

図 *253*：詳細ビューで右にスクロールした後



詳細ビューからのフロー検索リンク。

近隣の探索

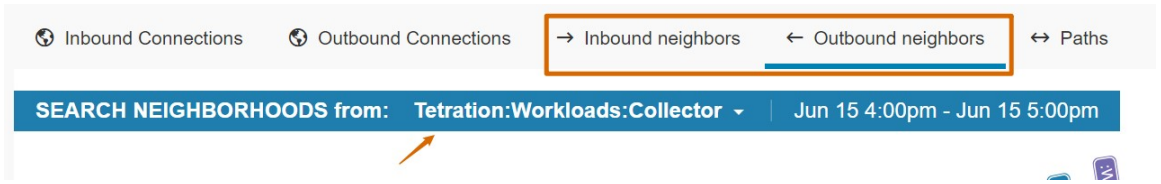
集約されたノード（範囲、フィルタ、クラスタ）データの調査には3つのバージョンがあります。

1. インバウンド：選択されたノードを宛先とする集約されたフロー
2. アウトバウンド：選択されたノードを送信元とする集約されたフロー
3. パス：1 つの送信元ノードと 1 つの宛先ノードが制約されているフローの、集約されたビューこれらはノードツーノードのエッジを集約したものですが、それ以外では無関係であることに注意してください。

インバウンドおよびアウトバウンドの探索

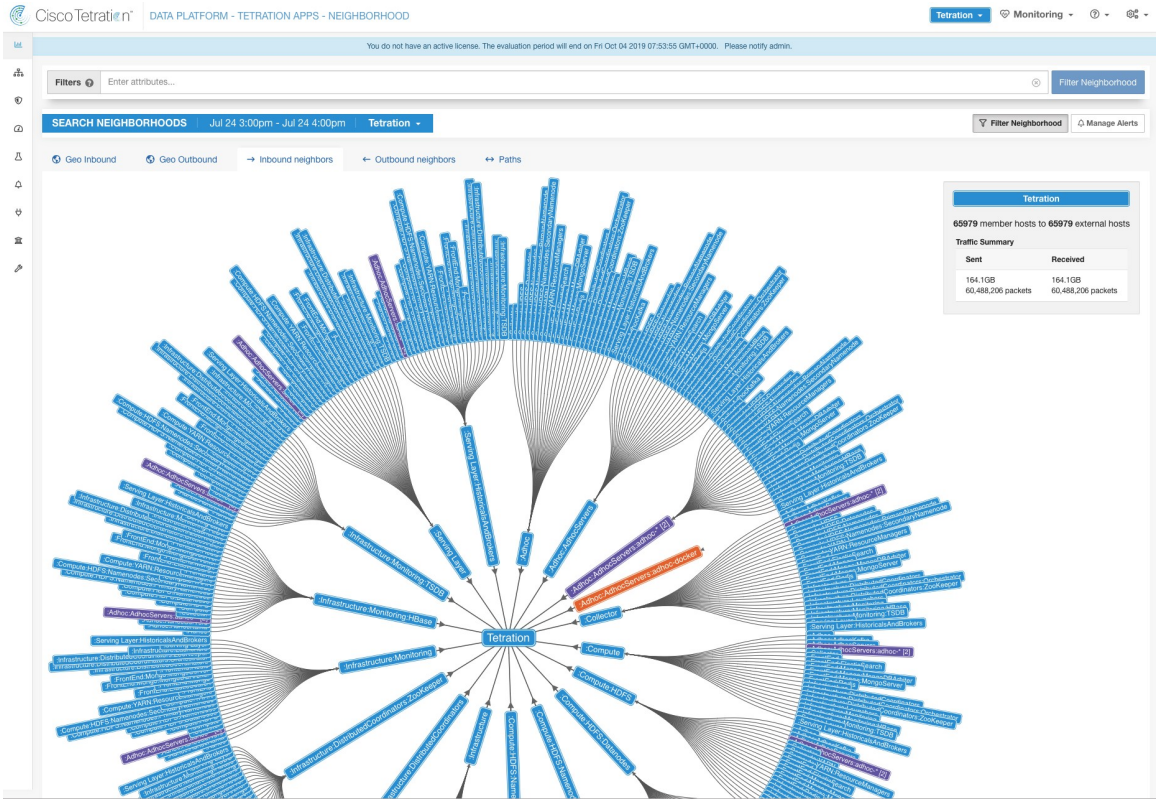
目的のノードを入力することを選択します。次に、インバウンドまたはアウトバウンドのいずれかを選択します

図 *254*：近隣データの探索



選択されたノードを中心に放射状のツリーが表示され、最大2ホップ離れた隣接ノードが内側に放射状に広がります。放射状ツリーの下には、選択されたノードへのパスのリストが表示されます。

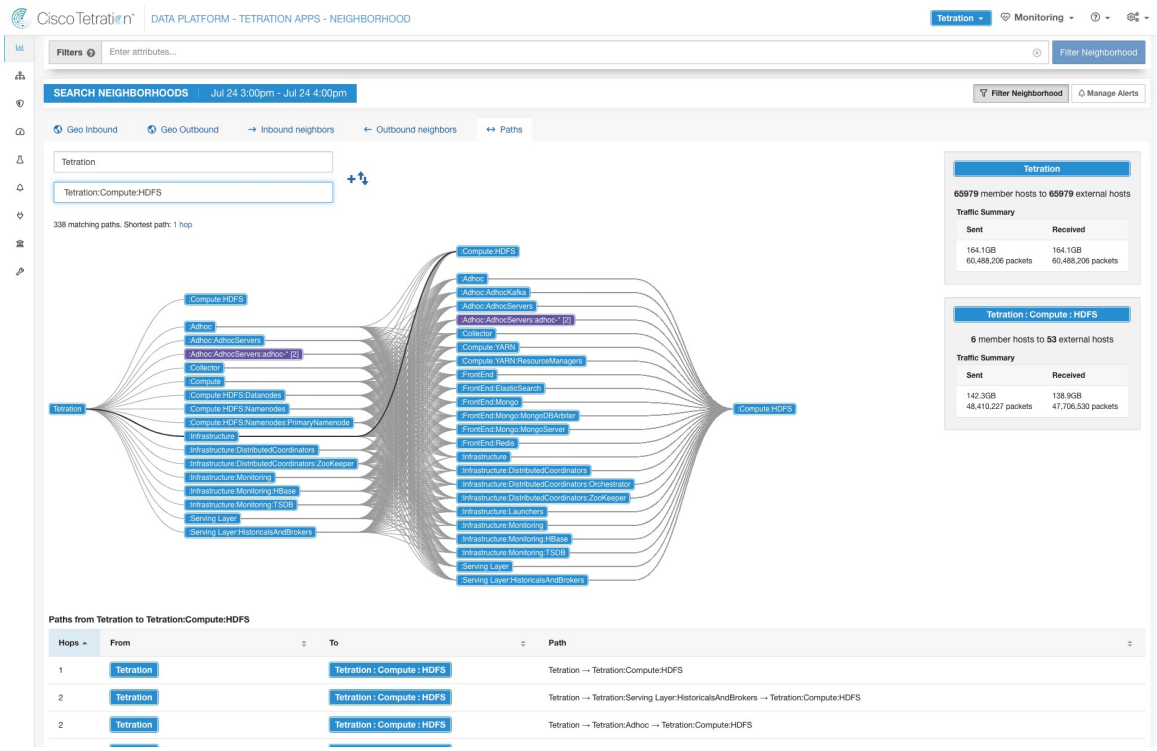
図 *255* : ノード



パスの探索

インバウンドまたはアウトバウンドの代わりに「パス」を選択すると、送信元と宛先の両方を指定できます。

図 *256 :* パス



フィルター オプション（Filter Options）

追加のフィルタオプションを指定することで、近隣グラフをフィルタ処理できます。現在サポートされているフィルタは、プロバイダーポートとプロトコルです。

図 *257* : ノードのフィルタリング

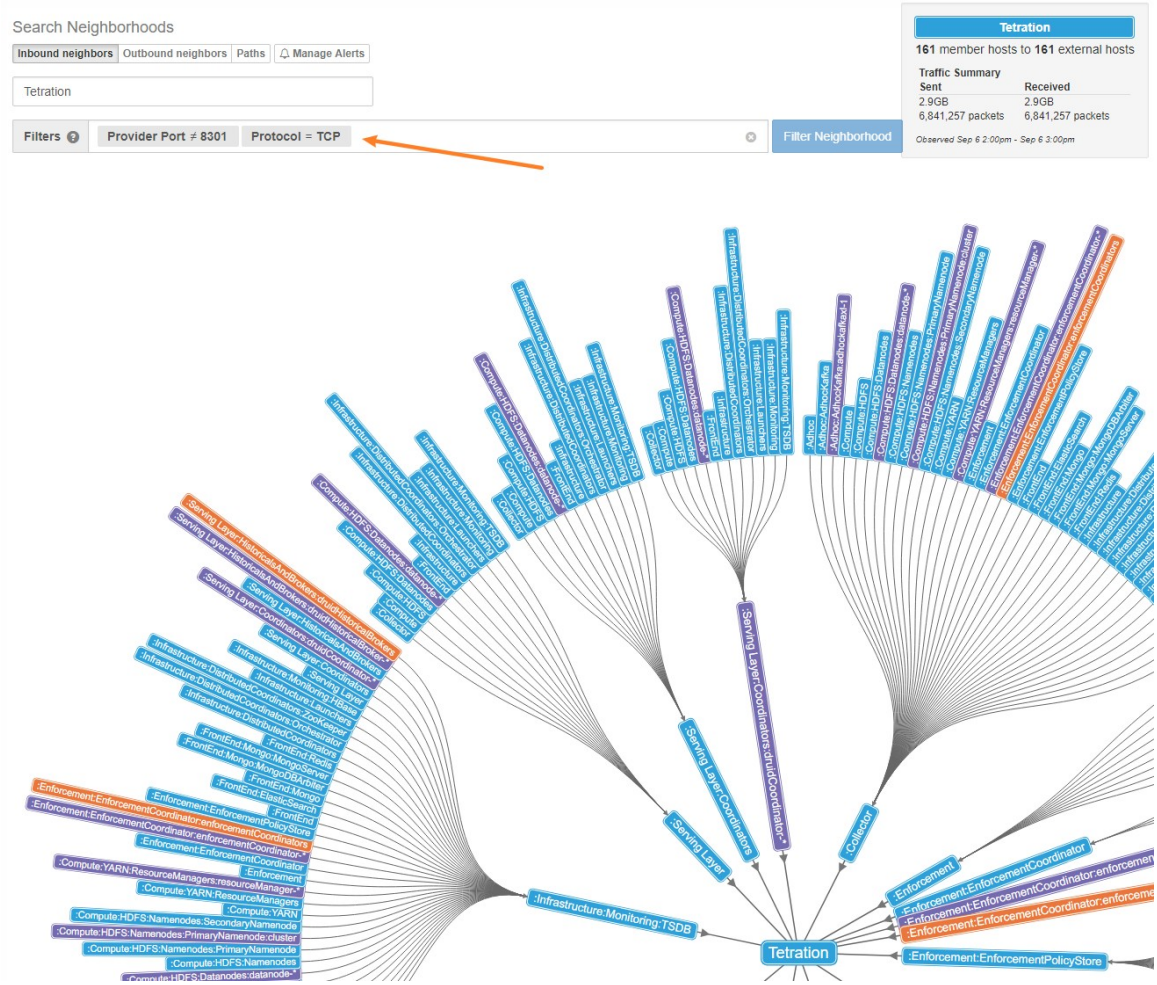
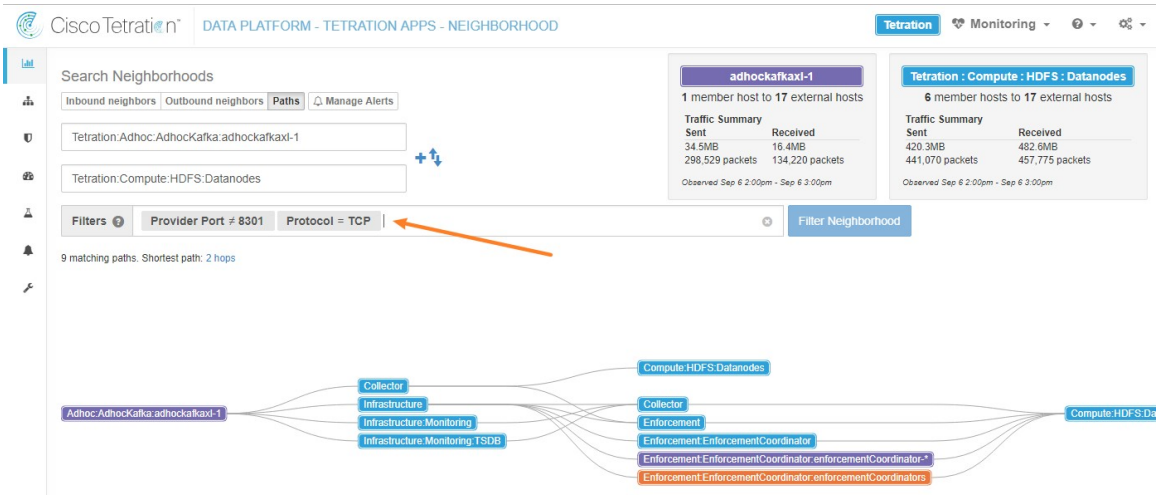
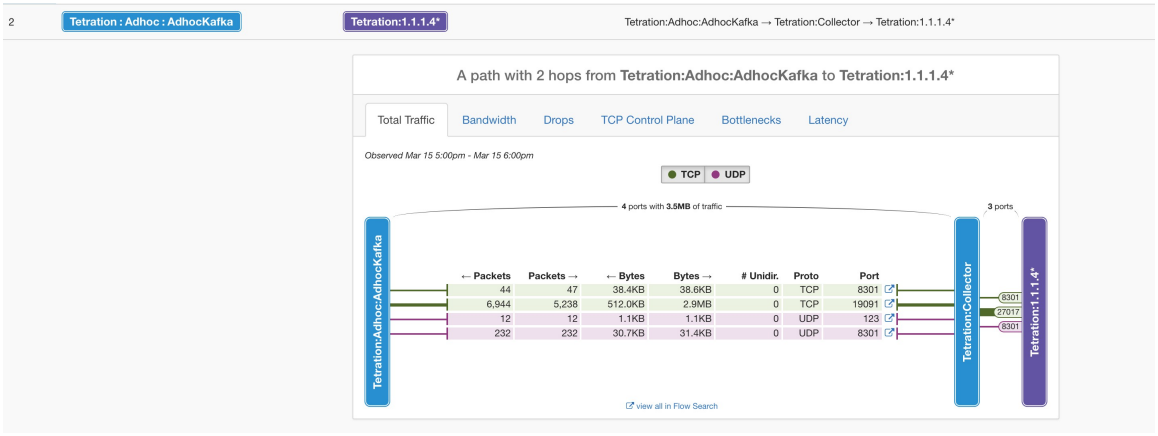


図 *258* : パスのフィルタリング



グラフの下にリストされているパスのいずれかをクリックすると、パスに関する詳細が展開され、フロー検索へのリンクが提供されます。

図 *259* : パスの詳細

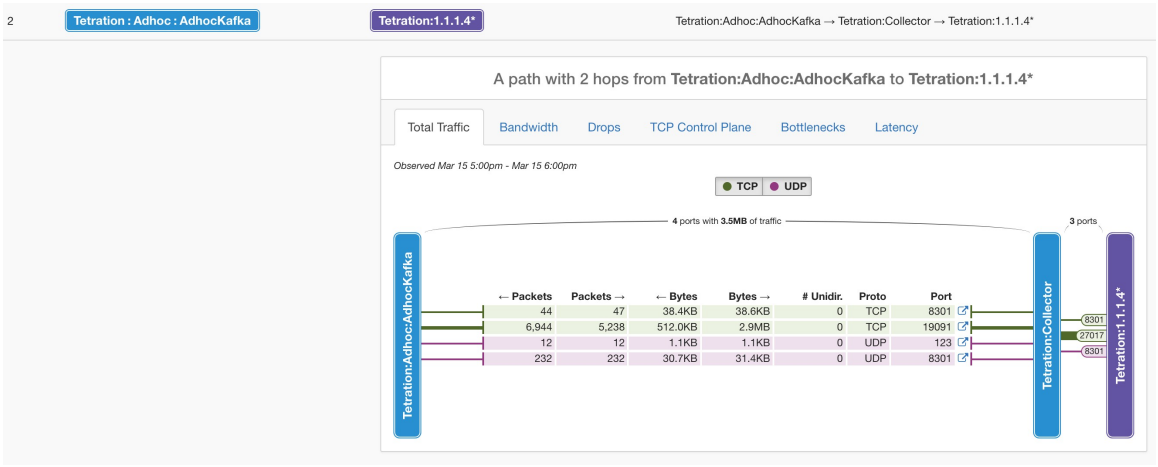


パスの詳細

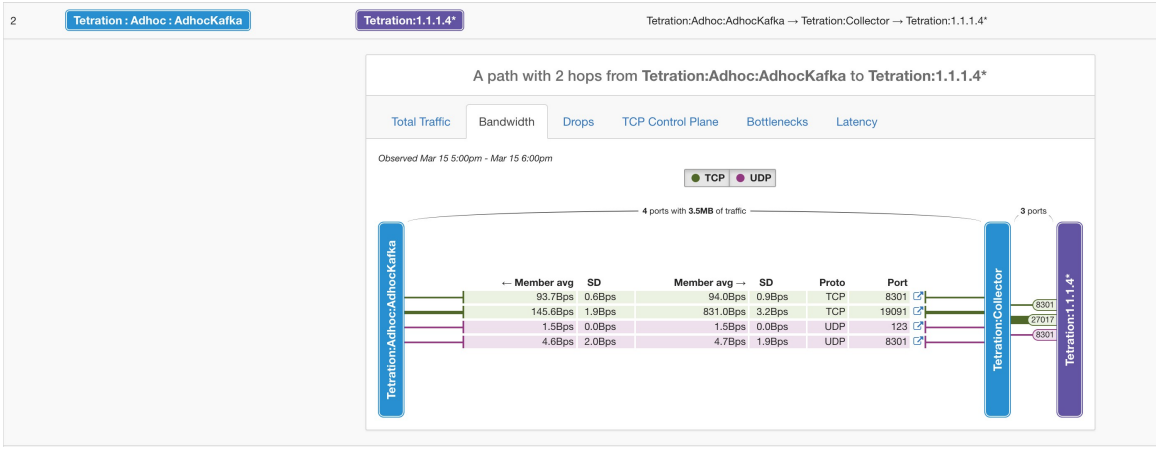
パスの詳細には、総トラフィック、帯域幅、ドロップ、TCP コントロールプレーン、ボトルネック、遅延など、さまざまなメトリックのグループを示すタブが含まれています。

（注） フローデータの収集メソッドによっては、一部のメトリックを使用できない場合があります。

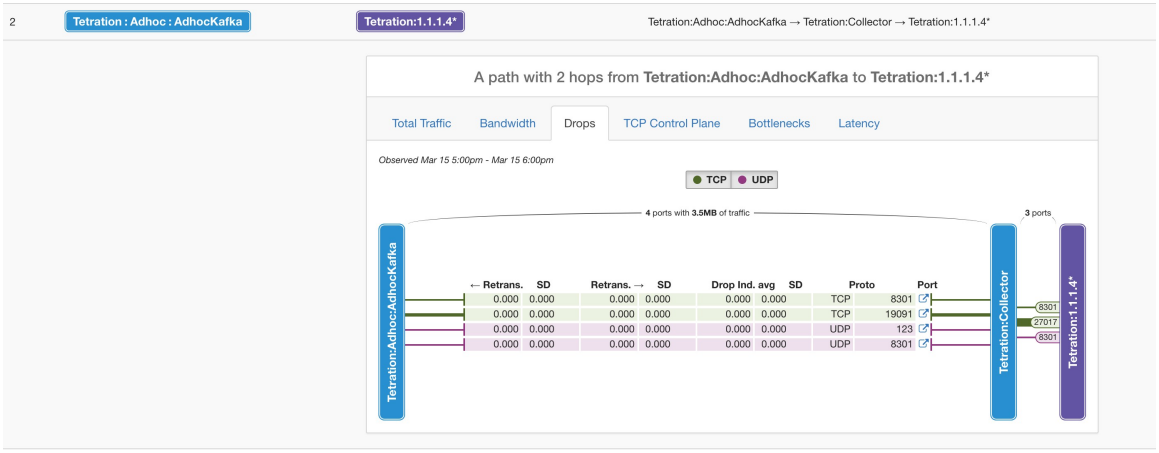
合計トラフィック



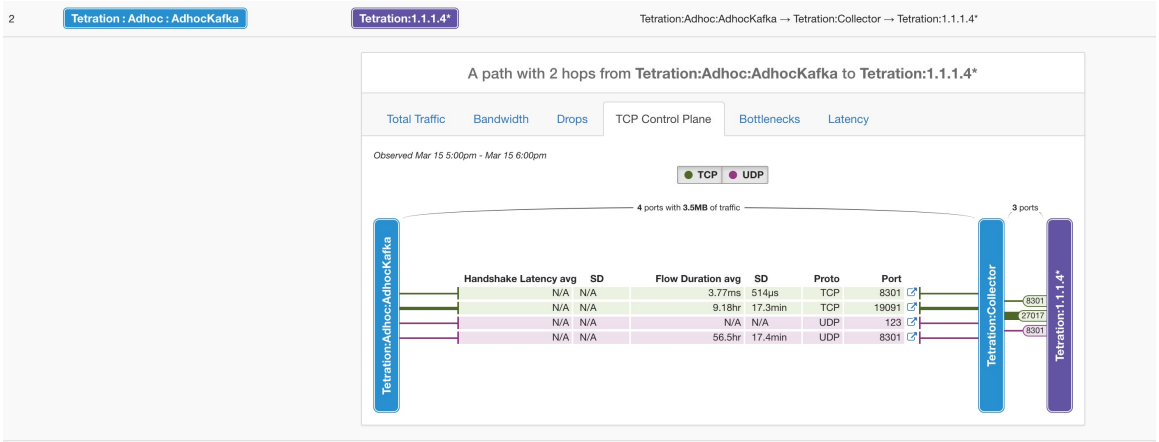
Bandwidth



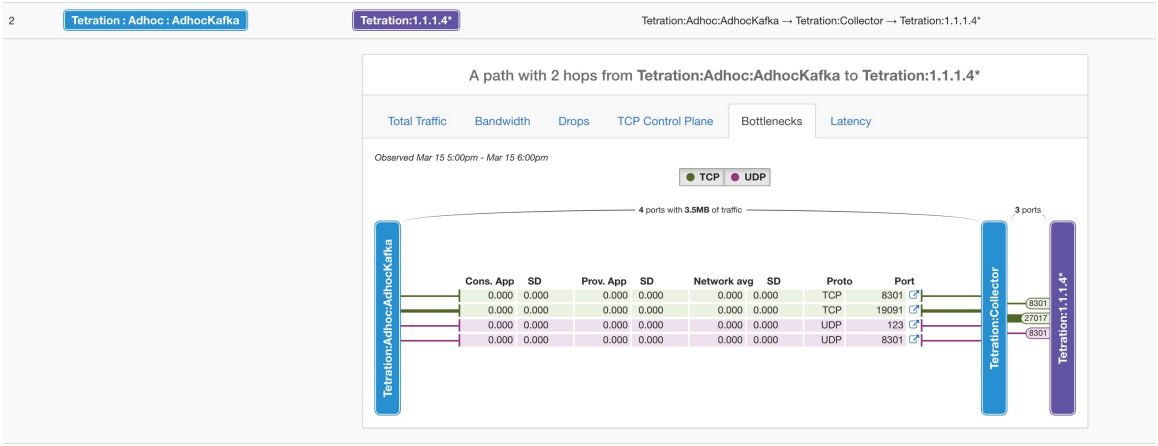
Drops



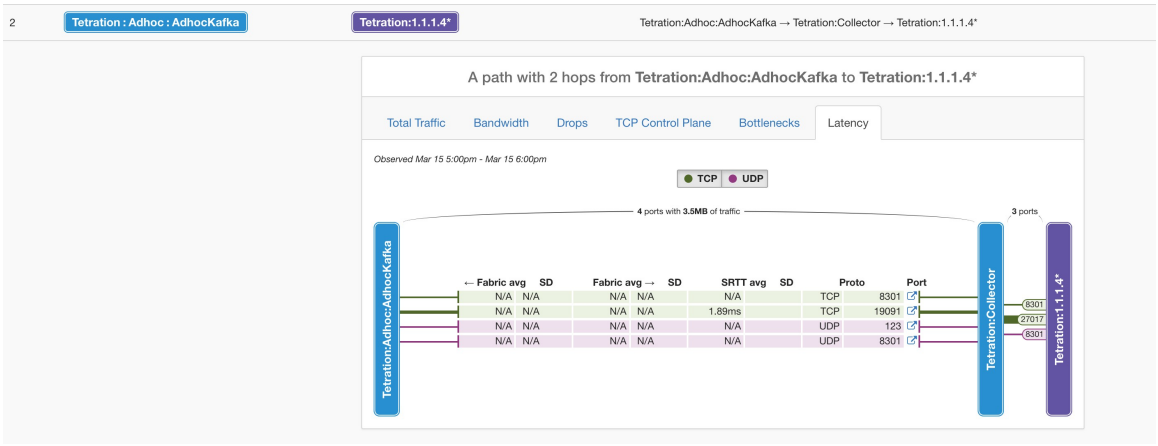
TCP コントロールプレーン



ボトルネック



遅延（Latency）



近接アラート

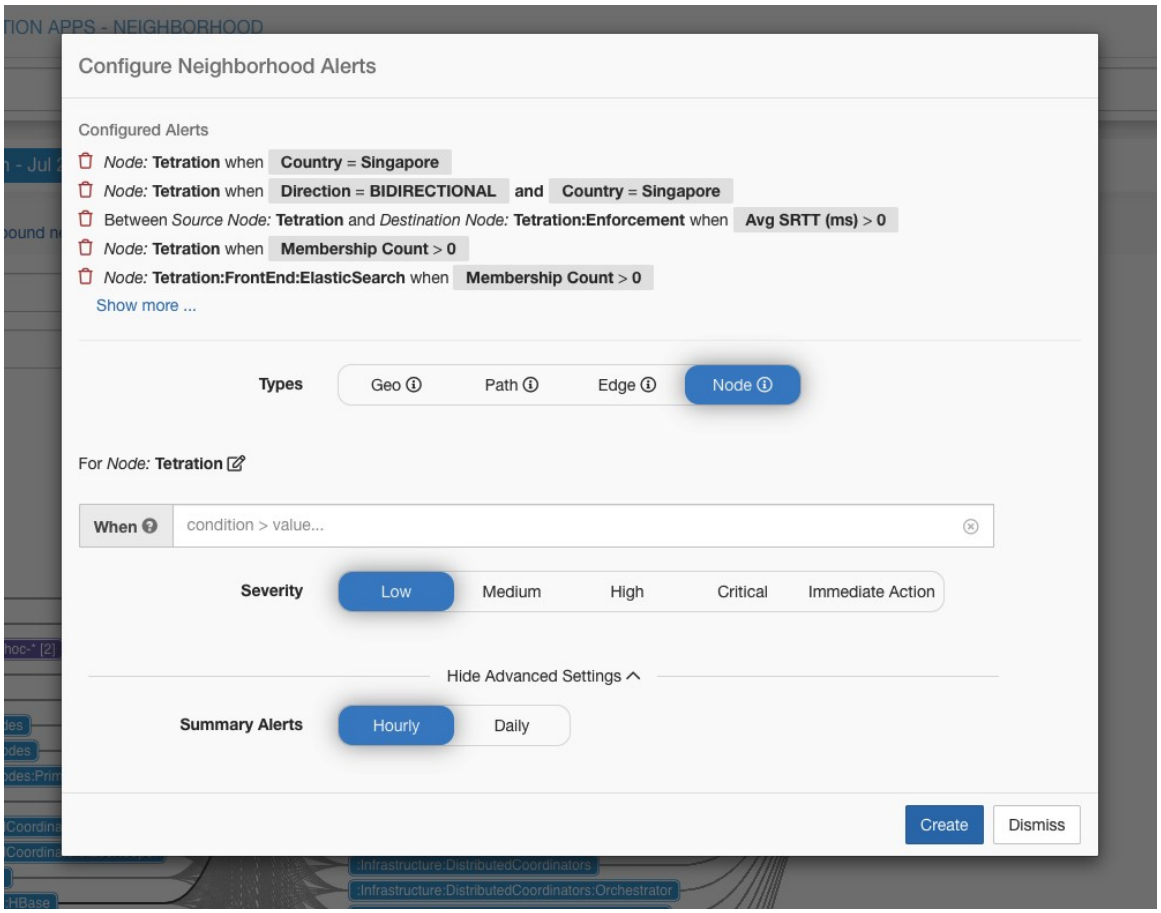
アラートの設定方法

- アラートを設定するには、[アラートの管理（Manage Alerts）] ボタンをクリックします。ボタンをクリックすると、アラート設定モーダルが開きます。ノード、エッジ、およびパスには、さまざまなタイプのアラートが用意されています。
- それぞれで使用可能なアラートトリガー設定を確認するには、ユーザーがタイプ（ノードなど）を選択し、クリックしてオプションを表示します。
- アラートトリガーの作成後、ユーザーはアラート設定を展開してアラートの頻度を変更できます。デフォルトの頻度は「毎時」ですが、「毎日」にも変更できます。

サポートされているアラート

タイプ	条件	コメント
地域	方向（Direction）	** ASO および国と組み合わせて使用する場合のみ
地域	ASO	方向（上）に応じて ASO の条件（= または≠）を確認
地域	国（Country）	方向（上）に応じて国の条件（= または/=）を確認
パス	任意のホップ	指定されたノードを経由しないパスをチェック
パス	パス	パスサイズを指定された値と比較
Edge	平均 SRTT	平均 SRTT を指定された値と比較
Edge	最大 SRTT	最大 SRTT を指定された値と比較
Edge	単方向のフロー	単方向のフローかどうかをチェック
ノード	メンバーシップの数	メンバーシップの数を指定された値と比較
ノード	隣接数	隣接数を指定された値と比較

図 *260 :* アラートの管理



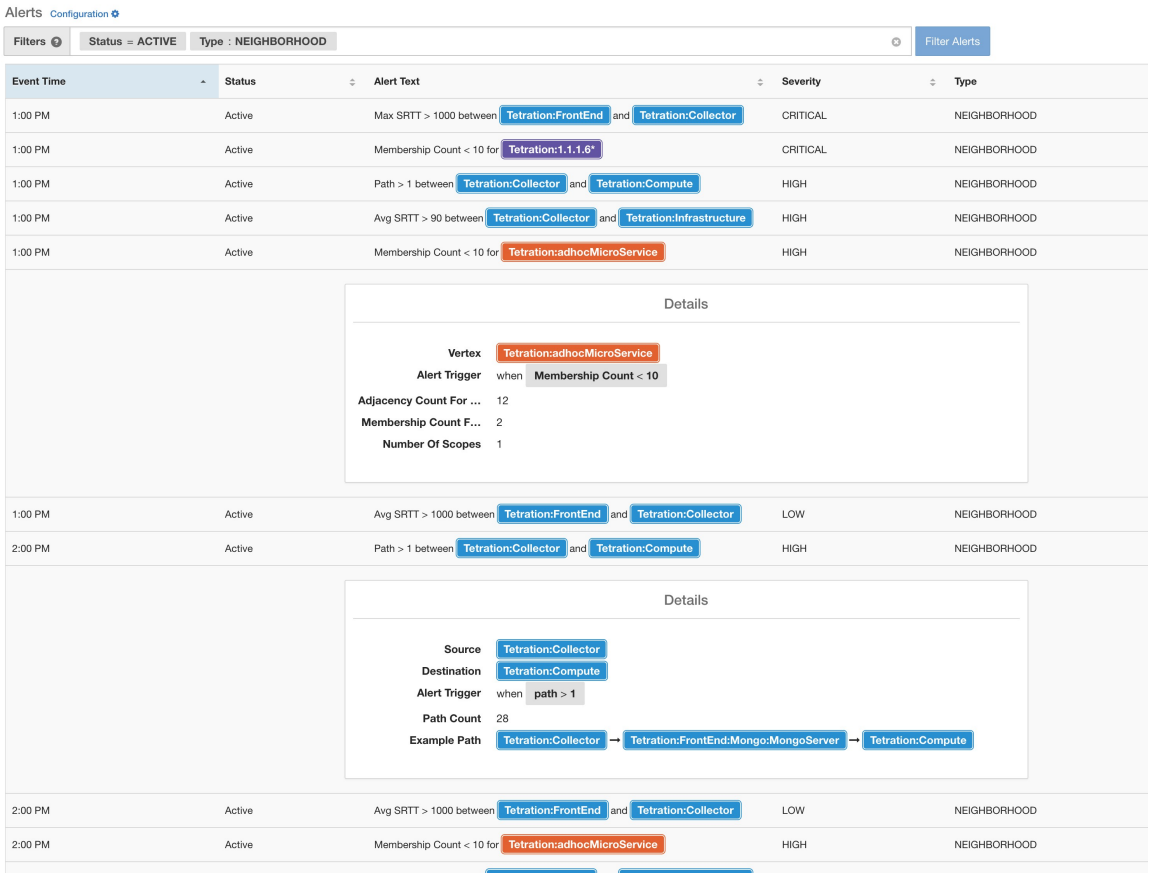
警告 サブ範囲またはフィルタが削除されても、サブ範囲またはフィルタで設定されたアラートは自動的に削除されません。同等のクエリを持つ新しいクラスタの関連性は維持されますが、クラスタまたはフィルタが最新のライブ分析ポリシーで使用されなくなった場合、それらのクラスタとフィルタを使用するアラートは生成されず、古いアラート設定が残ります。設定されたアラートを定期的に見直して、関連性を維持していることを確認する必要があります。

アラートの表示方法

- 近接アラートに対して有効な DataTap を選択する必要があります。アラートは、成功した場合のみ UI に表示されます。
  - アラートパブリッシャと通知者は、[アラート（Alerts）] → [構成（Configuration）]（[ルート範囲の所有者（Root Scope Owners）] または [サイト管理者（Site Admins）]）から選択できます。
- アラートを設定し、DataTap を設定すると、アラートは、[アラート（Alerts）] → [現在のアラート（Current Alerts）] の UI で確認できます。

- ユーザーは、フィルタ選択ボックスで **Type = NEIGHBORHOOD** を使用できます。その他のフィルタリングオプションについては、「現在のアラート」を参照してください。
- アラートをクリックすると、アラートの詳細が表示されます。

図 *261 :* 近接アラート



アラート詳細

一般的なアラート構造とフィールドに関する情報については、「一般的なアラート構造」を参照してください。alert_details フィールドは構造化されており、近隣アラートの次のサブフィールドが含まれています。

（注） サブジェクト（ノードの間隔名）は、アラートをトリガーした近隣ノードです。

フィールド	アラートタイプ	フォーマット	説明
neighborhood_subjects_id	*all*	string	近隣ノード ID

フィールド	アラートタイプ	フォーマット	説明
neighborhood_subjects_name	*all*	string	近隣ノード名
internal_trigger	*all*	処理のために	アラートトリガーを説明するクエリ（詳細は次の表を参照）
country	*geo*	string	国名
subdivision	*geo*	string	区画名
aso	*geo*	string	組織名
flow	*geo*	string	アラートをトリガーしたフローの詳細（src および dst ip）
vertex_neighborhood_subjects_id	*all*	string	neighbor_subjects_id と同じ
adjacency_count_for_example_vertex	ノード	integer	指定されたノードの隣接カウント
membership_count_for_example_vertex	ノード	integer	指定されたノードのメンバーシップ数
src_neighborhood_subjects_id	*edge*、*path*	string	送信元近隣のサブジェクトID（範囲、クラスタ、またはフィルタ）
src_neighborhood_subjects_name	*edge*、*path*	string	送信元近隣のサブジェクト名（範囲、クラスタ、またはフィルタ）
dst_neighborhood_subjects_id	*edge*、*path*	string	宛先近隣のサブジェクト ID（範囲、クラスタ、またはフィルタ）
dst_neighborhood_subjects_name	*edge*、*path*	string	宛先近隣のサブジェクト名（範囲、クラスタ、またはフィルタ）
number_of_edges	エッジ	integer	アラートがトリガーされたエッジの数
max_srtt	エッジ	integer	条件がトリガーされたフロー全体の SRTT の最大値

フィールド	アラートタイプ	フォーマット	説明
avg_srtt	エッジ	integer	トリガーされたアラートのフロー SRTT の平均値
unidirectional_flow_count	エッジ	string	フロー数（複数）
example_path_neighborhood_subjects_id	*path*	array[string]	トリガー条件に一致する 1 つのサンプルパスを構成する範囲、クラスタ、またはフィルタで構成される ID のリスト
example_path_neighborhood_subjects_name	*path*	array[string]	トリガー条件に一致する 1 つのサンプルパスを構成する範囲、クラスタ、またはフィルタで構成されるサブジェクトのリスト
number_of_unique_paths	*path*	integer	トリガー条件に一致する一意のパスの数

internal_trigger フィールドは構造化されており、アラートトリガーの次のサブフィールドが含まれています。

フィールド	フォーマット	説明
datasource	string	アラートタイプ
rules	string	クエリ評価ルールのコレクション
filters	string	組み合わせクエリルールのリスト
type	string	クエリルールタイプ（eq、lt、gt など. .）
値	string	アラート設定のユーザー入力値
label	string	「アラートのトリガー」

Geo（ASO）アラートの alert_details の例


```
{
  "neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2",
  "country":"United States",
  "subdivision":"Texas",
  "internal_trigger":{
    "datasource":"geo",
    "rules":{
       "filters":[
          {
             "field":"direction",
             "type":"eq",
             "value":"BIDIRECTIONAL"
          },
          {
             "field":"aso",
             "type":"eq",
             "value":"CISCOSYSTEMS"
          }
       ],
       "type":"and"
    },
    "label":"Alert Trigger"
 },
 "neighborhood_subjects_name":"Default",
 "vertex_neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2",
 "flow":"72.163.32.44 -> Default"
}
```


Geo（国）アラートの alert_details の例


```
{
  "neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2",
  "country":"Netherlands",
  "subdivision":"Provincie Flevoland",
  "internal_trigger":{
    "datasource":"geo",
    "rules":{
       "field":"country",
       "type":"eq",
       "value":"Netherlands"
    },
    "label":"Alert Trigger"
  },
  "neighborhood_subjects_name":"Default",
  "vertex_neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2",
  "flow":"173.38.201.67 -> Default"
}
```


ノード（隣接数）アラートの alert_details の例


```
{
  "adjacency_count_for_example_vertex":7,
  "neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2:c_5f04b0efc5445388852786b6",
  "internal_trigger":{
    "datasource":"vertex",
    "rules":{
       "field":"adjacency_count",
       "type":"gt",
       "value":-1
```

```
    },
    "label":"Alert Trigger"
 },
 "neighborhood_subjects_name":"Default:cluster",
"vertex_neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2:c_5f04b0efc5445388852786b6"

}
```

ノード（メンバーシップ数）アラートの alert_details の例

```
{
  "neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2",
  "internal_trigger":{
    "datasource":"vertex",
    "rules":{
       "field":"membership_count",
       "type":"gt",
       "value":0
    },
    "label":"Alert Trigger"
  },
  "neighborhood_subjects_name":"Default",
  "membership_count_for_example_vertex":156,
  "vertex_neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2"
}
```

Edge (srtt avg) アラートの alert_details の例

```
{
  "internal_trigger":{
    "datasource":"edge",
    "rules":{
       "field":"srtt_usec_avg",
       "type":"gt",
       "value":-1
    },
    "label":"Alert Trigger"
  },
  "src_neighborhood_subjects_id":"5efcfe0f497d4f49adebc74e",
  "dst_neighborhood_subjects_name":"Tetration:AdhocKafka",
  "dst_neighborhood_subjects_id":"5efcfe0f497d4f49adebc6ee",
  "number_of_edges":2,
  "max_srtt":0,
  "avg_srtt":0,
  "src_neighborhood_subjects_name":"Tetration:Collector"
}
```

Edge（max srtt）アラートの alert_details の例

```
{
  "internal_trigger":{
    "datasource":"edge",
    "rules":{
       "field":"srtt_usec_max",
       "type":"gt",
       "value":-1
    },
    "label":"Alert Trigger"
  },
  "src_neighborhood_subjects_id":"5efcfe0f497d4f49adebc74e",
```

```
  "dst_neighborhood_subjects_name":"Tetration:AdhocKafka",
  "dst_neighborhood_subjects_id":"5efcfe0f497d4f49adebc6ee",
  "number_of_edges":2,
  "max_srtt":0,
  "avg_srtt":0,
  "src_neighborhood_subjects_name":"Tetration:Collector"
}
```


Edge（一方向フロー）アラートの alert_details の例


```
{
  "unidirectional_flow_count":1,
  "internal_trigger":{
    "datasource":"edge",
    "rules":{
       "field":"num_unidirectional_flows",
       "type":"gt",
       "value":0
    },
    "label":"Alert Trigger"
  },
  "src_neighborhood_subjects_id":"5efcfe0f497d4f49adebc74e",
  "dst_neighborhood_subjects_name":"Tetration:AdhocKafka",
  "dst_neighborhood_subjects_id":"5efcfe0f497d4f49adebc6ee",
  "number_of_edges":1,
  "src_neighborhood_subjects_name":"Tetration:Collector"
}
```


パス（指定された 2 つのノード間のホップサイズ）アラートの alert_details の例

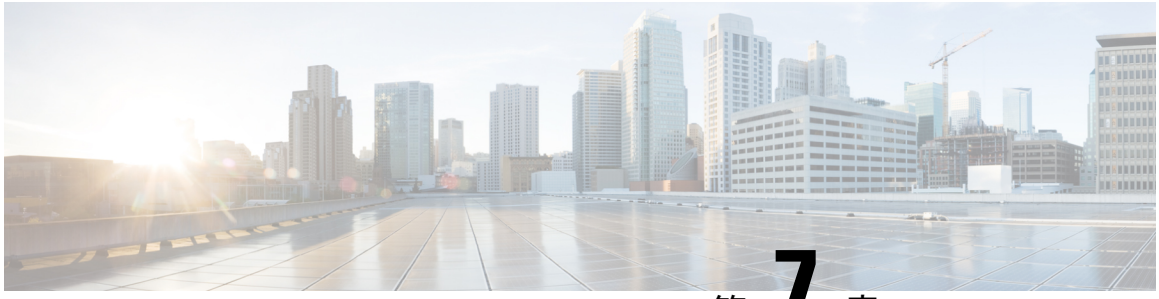

```
{
  "number_of_unique_paths":2,
  "example_path_neighborhood_subjects_id":[
    "5efcfdf5497d4f474f1707c2",
    "5efcfdf5497d4f474f1707c2:c_5f04b0efc5445388852786b6",
    "5efcfdf5497d4f474f1707c2:c_5f04b0efc5445388852786b7"
  ],
  "internal_trigger":{
    "datasource":"hop",
    "rules":{
       "field":"hops",
       "type":"gt",
       "value":0
    },
    "label":"Alert Trigger"
  },
  "src_neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2",
  "dst_neighborhood_subjects_name":"Default:collectorDatamover-*",
 "dst_neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2:c_5f04b0efc5445388852786b7",

  "src_neighborhood_subjects_name":"Default",
  "example_path_neighborhood_subjects_name":[
    "Default",
    "Default:cluster",
    "Default:collectorDatamover-*"
  ]
}
```

パス（指定されたノードを経由しない任意のホップ）アラートの alert_details の例


```
{
"number_of_unique_paths":2,
"example_path_neighborhood_subjects_id":[
"5efcfdf5497d4f474f1707c2",
"5efcfdf5497d4f474f1707c2:c_5f04b0efc5445388852786b6",
"5efcfdf5497d4f474f1707c2:c_5f04b0efc5445388852786b7"
],
"internal_trigger":{
"datasource":"hop",
"rules":{
"filter":{
"field":"path_by_neighborhood_subjects_id",
"type":"contains",
"value":"5efcfdf5497d4f474f1707c2:c_5f04b0efc5445388852786b5"
},
"type":"not"
},
"label":"Alert Trigger"
},
"src_neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2",
"dst_neighborhood_subjects_name":"Default:collectorDatamover-*",
"dst_neighborhood_subjects_id":"5efcfdf5497d4f474f1707c2:c_5f04b0efc5445388852786b7
‹→",
"src_neighborhood_subjects_name":"Default",
"example_path_neighborhood_subjects_name":[
"Default",
"Default:cluster",
"Default:collectorDatamover-*"
]
}
```

パス（指定されたノードを経由しない任意のホップ）アラートの **alert_details** の例



第 7 章

ポリシー

- セグメンテーションポリシーの基本 （513 ページ）
- ワークスペースを使用したポリシーの管理 （514 ページ）
- ポリシーについて （521 ページ）
- ポリシーの作成と検出 （524 ページ）
- ワークロードのグループ化：クラスタとインベントリフィルタ （595 ページ）
- ポリシーの複雑さの対処 （607 ページ）
- ポリシーの削除について （632 ページ）
- ポリシーの確認と分析 （633 ページ）
- ポリシーの適用 （654 ページ）
- 適用されたポリシーの変更 （669 ページ）
- ポリシーバージョン（v* および p*）について （673 ページ）
- カンバセーション （679 ページ）
- 自動ポリシー検出用の自動ロードバランサ設定（F5 のみ） （687 ページ）
- ポリシーパブリッシャ （692 ページ）

セグメンテーションポリシーの基本

セグメンテーションポリシーとマイクロセグメンテーションポリシーの目的は、組織がビジネスを遂行するために必要なトラフィックのみを許可し、他のすべてのトラフィックをブロックすることです。事業運営を中断することなく、ネットワークの攻撃対象領域を縮小することが目標です。

Cisco Secure Workload のセグメンテーションポリシーは、送信元、宛先、ポート、プロトコル、および通常はプラットフォーム固有のその他のいくつかの属性に基づいて、トラフィックを許可またはブロックします。

一部のポリシーを手動で作成し、Cisco Secure Workload の強力な自動ポリシー検出機能を使用して、既存のネットワークトラフィックに基づいて他のポリシーを生成します。

ポリシーを確認、改良、分析し、組織が必要とするトラフィックのみを許可することが確実になったら、ポリシーを適用します。

重要 マイクロセグメンテーションは、基本的には、各ワークロードの周囲にファイアウォールを作成します。

そのため、トラフィックがコンシューマとプロバイダーの各ペア間を通過するには、カンバセーションの両端でカンバセーションの発生が許可されている必要があります。つまり、コンシューマとプロバイダーのそれぞれに、トラフィックを許可するポリシーが必要です。

（注） ファイアウォールルール、エッジ、およびクラスタエッジという用語が、「ポリシー」を意味するために使用されることがあります。

ワークスペースを使用したポリシーの管理

ワークスペース（以前は「アプリケーションワークスペース」または「アプリケーション」と呼ばれていました）は、ポリシーを使用および管理する場所です。

その範囲に関連付けられたワークスペース内のポリシーの作成、分析、適用など、特定の範囲のポリシー関連のすべてのアクティビティを実行します。

各ワークスペースは隔離された環境を提供するため、他のワークスペースに影響を与えずに実験を行うことができます。

ワークスペースへのユーザーアクセスの制御

ワークスペースは、同じチームの複数のユーザーが共有ドキュメントとして使用するためのものです。

ワークスペースへのアクセスを制御するには、ワークスペースに関連付けられた範囲のユーザーロールを割り当てます。ロール （1023 ページ） を参照してください。

ポリシーの操作：[ワークスペース（Workspaces）] ページへの移動

- **ポリシーを操作したり、既存のアプリケーション ワークスペースを表示したり、新しいワークスペースを作成したりするには、次の手順を実行します。**

  ウィンドウの左側にあるナビゲーションバーから **[防御（Defend）]** > **[セグメンテーション（Segmentation）]** を選択します。

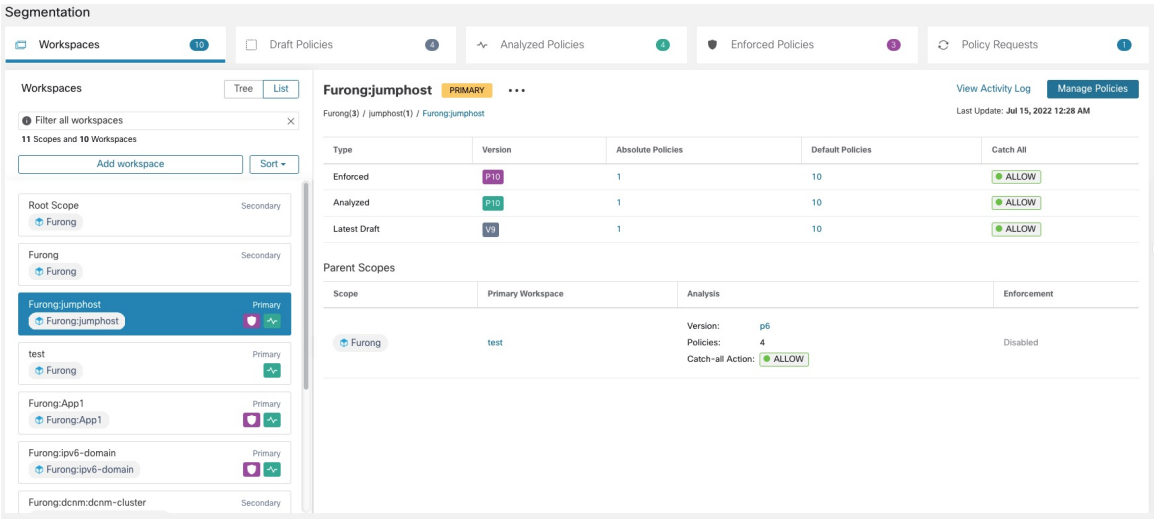
- **特定のワークスペースを表示するには、次の手順を実行します。**

  [ワークスペース（Workspaces）] ページの左側にある範囲のリストで、ワークスペースに関連付けられている範囲に移動し、ワークスペースをクリックします。現在アクティブなワークスペースがリストで強調表示されます。

- ワークスペースを表示しているときに、ワークスペースのリストに戻るには、次の手順を実行します。

  表示しているページの左側付近にある [ワークスペース（Workspaces）] リンクをクリックします。

図 *262 :* ワークスペース管理ページ



ワークスペースの作成

範囲のポリシーを作成するには、まずその範囲のワークスペースを作成します。

ワークスペースを作成するには、次の手順を実行します。

1. ウィンドウの左側のナビゲーションメニューから、**[防御（Defend）]**>**[セグメンテーション（Segmentation）]** を選択します。
2. ページの左側にある範囲のリストで、ポリシーを作成する範囲を検索するか、その範囲までスクロールします。
3. 範囲にカーソルを合わせ、青いプラス記号が表示されたらクリックします。
4. フォームに入力し、完了したら [作成（Create）] をクリックします。

範囲のワークスペースがすでに存在する場合、追加のワークスペースは自動的にセカンダリワークスペースとして作成されます。

プライマリおよびセカンダリワークスペース

範囲ごとに、1 つのプライマリワークスペースと複数のセカンダリワークスペースを作成できます。

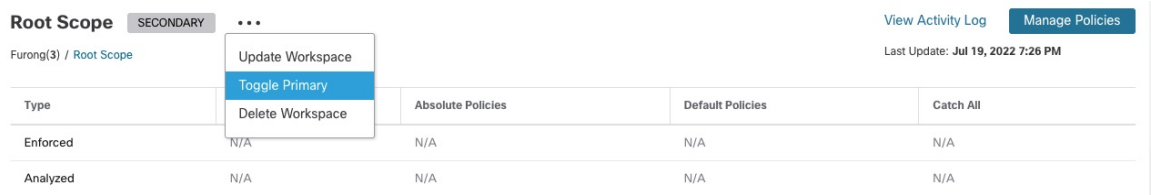
適用は、プライマリワークスペースのみで可能です。プライマリワークスペースでのみ使用できるその他の機能には、コンシューマとプロバイダーが異なる範囲に存在するポリシーを管理する機能、ライブポリシー分析、コンプライアンスレポート、およびコラボレーティブセキュリティ ポリシー定義があります。

プライマリワークスペースの既存のポリシーを保持する場合は、セカンダリワークスペースを使用してポリシーを試します。

ワークスペースをプライマリまたはセカンダリに変更するには、次の手順を実行します。

ページ上部のワークスペース名の横にあるメニューアイコンをクリックし、[プライマリの切り替え（Toggle Primary）] を選択することで、いつでもワークスペースをプライマリからセカンダリに、またはその逆に切り替えることができます。

図 *263 :* プライマリとセカンダリの間でワークスペースを切り替える



ワークスペース名の変更

ワークスペース名を変更するには、次の手順を実行します。

ページの上部付近に表示されるワークスペースタイプ（[プライマリ（Primary）] または [セカンダリ（Secondary）]）の横にある ••• をクリックし、[ワークスペースの更新（Update Workspace）] を選択します。

範囲内のワークロードの表示

任意のワークスペースで、[一致するインベントリ（Matching Inventories）] タブをクリックします。

ワークスペース内の検索

ワークスペース内でワークロード、クラスタ、またはポリシーを検索するには、次の手順を実行します。

1. **[防御（Defend）] > [セグメンテーション（Segmentation）]** の順に選択します。
2. 左側の範囲のリストから、目的の範囲とワークスペースをクリックします。
3. [ポリシーの管理（Manage Policies）] をクリックします。
4. 虫眼鏡マークをクリックします。
5. 検索条件を入力します。

検索条件

複数の条件は、論理 AND として扱われます。

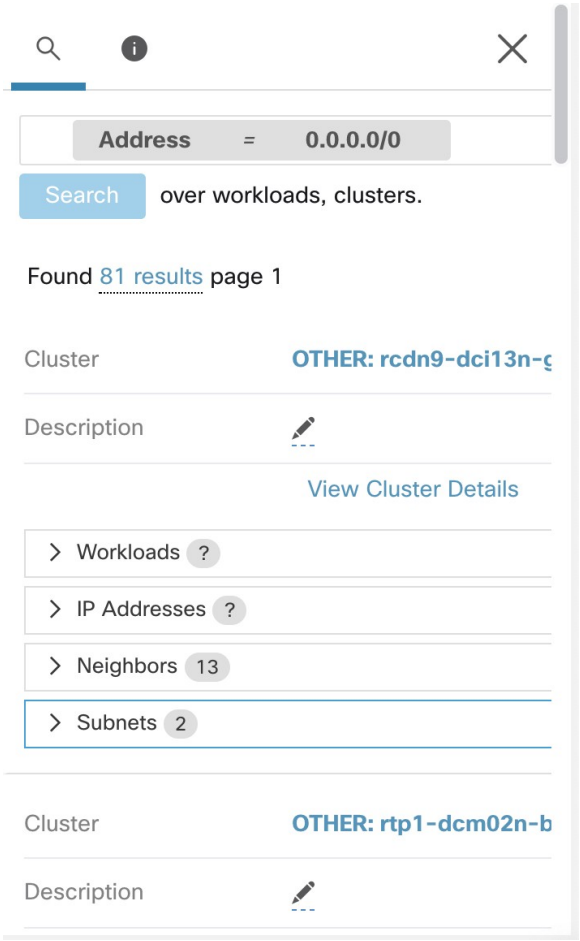
IP アドレスと数値の場合は、次のようになります。

- 'port: 80,443' のように、カンマを使用して論理 OR を指定します。
- 数値の範囲クエリ「port: 3000-3999」もサポートされています。

フィルタ	説明
名前	クラスタ名またはワークロード名を入力します。大文字と小文字を区別する部分文字列検索を実行します。
説明	クラスタの説明を検索します。
承認済み（Approved）	「true」または「false」の値を使用して、承認されたクラスタに一致します。
アドレス（Address）	CIDR 表記（例：10.11.12.0/24）を使用してサブネットまたは IP アドレスを入力します。このサブネットと重複するワークロードまたはクラスタに一致します。
スーパーネット（Supernet）	ワークロードがこのサブネットに完全に含まれているクラスタと一致するように、CIDR 表記（例：10.11.12.0/24）を使用してサブネットを入力します。
Process	大文字と小文字を区別する部分文字列検索を使用して、ワークロードプロセスを検索します。
プロセス UID（Process UID）	ワークロードプロセスのユーザー名を検索します。
ポート（Port）	ワークロードプロバイダーポートとポリシーポートの両方を検索します。
[Protocol]	ワークロード プロバイダー プロトコルとポリシープロトコルの両方を検索します。
コンシューマ名（Consumer Name）	ポリシーのコンシューマクラスタ名に一致します。大文字と小文字を区別する部分文字列の一致を実行します。
プロバイダー名（Provider Name）	ポリシーのプロバイダークラスタ名に一致します。大文字と小文字を区別する部分文字列の一致を実行します。
コンシューマアドレス（Consumer Address）	提供された IP またはサブネットとコンシューマアドレスが重複するポリシーに一致します。

フィルタ	説明
プロバイダーアドレス（Provider Address）	提供された IP またはサブネットとプロバイダー アドレスが重複するポリシーに一致します。

検索の例

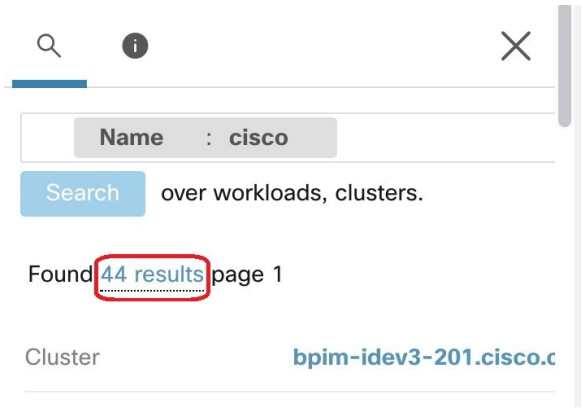


特定のタイプによる検索結果のフィルタリング

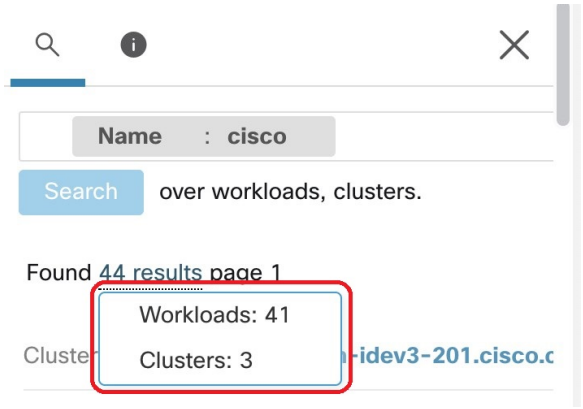
検索結果には、複数のタイプのオブジェクトが含まれている場合があります（ワークロードとクラスタなど）。

特定のタイプで検索結果をフィルタするには、次の手順を実行します。

1. 結果の合計をクリックします。



2. ドロップダウンからタイプを選択します。



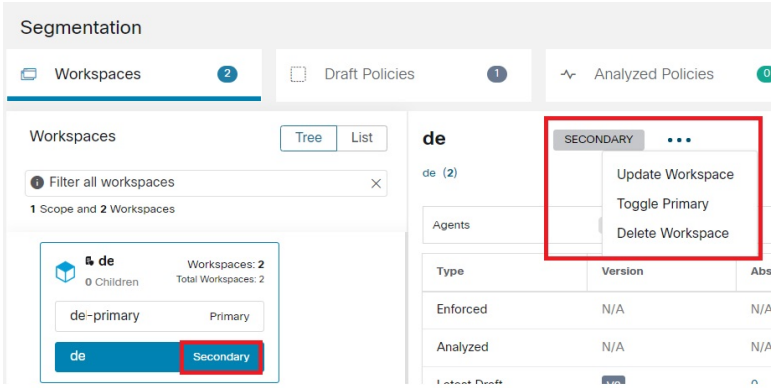
3. タイプフィルタが追加され、検索が再実行されます。

ワークスペースの削除

削除できるのは、セカンダリ（プライマリではない）ワークスペースのみです。ワークスペースをセカンダリに切り替えるには、プライマリおよびセカンダリワークスペース（515ページ）を参照してください。

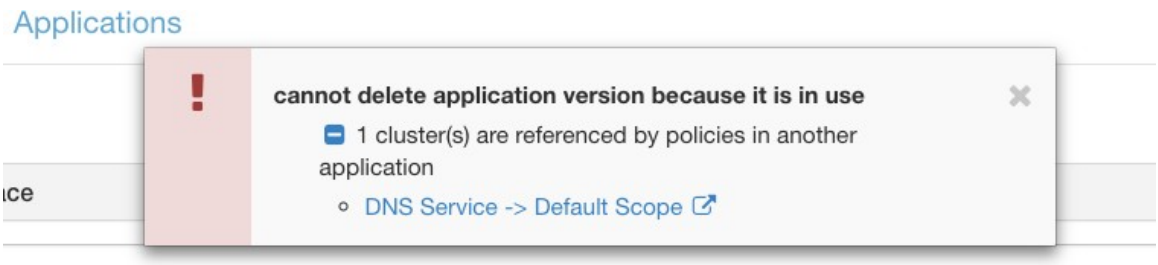
ワークスペースを削除するには、次の手順を実行します。

1. [防御（Defend）]>[セグメンテーション（Segmentation）]を選択します。
2. ページの左側にある範囲のリストで、削除するワークスペースが含まれている範囲に移動して、その範囲をクリックします。
3. 削除するワークスペースをクリックします。
4. [セカンダリ（Secondary）]の横にある •••をクリックし、[ワークスペースの削除（Delete Workspace）]を選択します。



ワークスペース内のワークロードまたはクラスタが、提供サービスの結果として別のワークスペース内のポリシーによって参照されている場合、依存ワークスペースは削除できず、依存関係のリストが返されます。この情報を使用して、依存関係を修正できます。

図 *264* : ワークスペースの削除を妨げる項目のリスト



まれに、ワークスペースAがワークスペースBのクラスタに依存し、ワークスペースBがワークスペースAのクラスタに依存する相互依存関係が発生することがあります。この場合、個々のポリシーや公開されたポリシーバージョン（p*）を削除する必要があります。「削除制限」エラーでは、すべてのポリシーへのリンクが表示されるため、この削除を実行できます。

p*バージョンを削除するには、分析されたポリシーバージョンの表示、比較、および管理（652ページ）または適用されたポリシーバージョンの表示、比較、および管理（669ページ）を参照してください。

ポリシーについて

ポリシー属性

表 *25 :* ポリシー プロパティ

セキュリティポリシーのプロパティ	説明
ポリシーが定義されている範囲	ポリシーは通常、ポリシーが定義されているワークスペースに関連付けられた範囲のメンバーであるワークロードにのみ影響します。 （ただし、ポリシーの複雑さの対処（607ページ）のトピックも参照してください）。 詳細については、「ポリシーの例（523ページ）」を参照してください。
コンシューマ	サービスのクライアントまたは接続のイニシエータ。 範囲、クラスタ、またはインベントリフィルタは、すべてポリシーのコンシューマとして使用できます。 ポリシーでのコンシューマとプロバイダーについて（523ページ）の重要な情報を参照してください。
プロバイダ	サーバーまたは接続の受信側。 範囲、クラスタ、またはインベントリフィルタは、すべてポリシーのプロバイダーとして使用できます。 ポリシーでのコンシューマとプロバイダーについて（523ページ）の重要な情報を参照してください。
プロトコルとポート[ぷろとこるとぽーと]	許可またはブロックする必要がある、プロバイダーによって利用可能になるサービスのサーバー（リスニング）ポートと IP プロトコル。
操作	許可または拒否：指定されたサービスのポートやプロトコルでのコンシューマからプロバイダーへのトラフィックを許可するかドロップするかを示します。
ランクと優先順位	ワークスペース内のポリシーのランクと優先順位の詳細については、ポリシーのランク：絶対、デフォルト、キャッチオール（522ページ）を参照してください。

ポリシーのランク：絶対、デフォルト、キャッチオール

ポリシーのランクは、優先順位リストの下位（または範囲ツリーの下位範囲）にあるより具体的なポリシーによってポリシーをオーバーライドできるかどうかを決定します。すべての範囲で最も優先順位の低いポリシーは、常にキャッチオールルールです。

ポリシーランク	説明
絶対値（Absolute）	絶対ポリシーは、ポリシーリストの下位の（したがって優先度の低い）アプリケーション固有のポリシーまたは範囲ツリーの下位の範囲で矛盾する場合でも有効です。一般に、絶対ポリシーを使用して、ベストプラクティスを適用したり、さまざまなゾーンを保護したり、特定のワークロードを検疫したりします。たとえば、絶対ポリシーを使用して、DNSまたはNTPサーバーへのトラフィックを制御したり、規制要件を満たしたりします。 絶対ポリシーは、ポリシー優先順位リストのデフォルトポリシーの上にリストされます。
デフォルト	デフォルトポリシーは、ポリシーリストの下位のポリシー、または範囲ツリーの下位の範囲のポリシーによってオーバーライドされます。一般に、非常にきめの細かいポリシーがデフォルトポリシーです。 デフォルトポリシーは、ポリシー優先順位リストの絶対ポリシーの下にリストされています。
Catch-All	各ワークスペースには、ワークスペースで明示的に指定されたすべてのポリシーと一致しない各方向のトラフィックを処理するキャッチオールポリシーがあります。Catch-Allアクションは、許可または拒否です。 一般に、Catch-Allポリシーは次のように設定します。 • 範囲ツリーの上位の範囲のトラフィックを**許可**し、ツリーの下位の範囲のポリシーがトラフィックを評価できるようにします。 • 範囲ツリーの下部にある最も限定的なリーフでトラフィックを**拒否**します。 これにより、ツリーのすべての範囲内のポリシーにトラフィックを照合する機会が与えられ、どの範囲内のポリシーにも一致しないトラフィックはブロックされます。 キャッチオールルールは、ワークスペース内の各ワークロードのすべてのインターフェイスに適用されます。

ポリシーの継承と範囲ツリー

ワークロードは階層型範囲ツリーに編成されているため、ツリーの上位またはその近くの範囲で、一般的なポリシーを一旦作成することができます。必要に応じて、そのポリシーをツリー内のその範囲の下にあるすべての範囲内のすべてのワークロードに適用することができます。

ツリーの下位にあるより具体的なポリシーによって、一般的なポリシーをオーバーライドできるかどうかを指定します。

ポリシーのランク：絶対、デフォルト、キャッチオール（522 ページ）を参照してください。

ポリシーでのコンシューマとプロバイダーについて

ポリシーで指定されたコンシューマとプロバイダーには、次の目的があります。

- ポリシーまたはファイアウォールルールを受け取るワークロードまたは Secure Workload エージェントを指定します。
- ワークロードにインストールされているファイアウォールルールが適用される IP アドレスのセットを指定します。

ホストに複数のインターフェイス（IP アドレス）がある場合、ポリシーはすべてのインターフェイスに適用されます。



重要 上記は、ワークロードでのファイアウォールルールのプログラミング方法のデフォルトの動作です。ファイアウォールルールで指定された IP アドレスが、ポリシーがインストールされているワークロードの IP アドレスと異なる場合は、ポリシーで 2 つの目的のコンシューマとプロバイダーを分ける必要がある場合があります。有効なコンシューマまたは有効なプロバイダー（629 ページ）を参照してください。

ポリシーの例

次のポリシーの例は、ポリシーが定義されている範囲の重要性、ポリシー継承の影響、および正確なポリシーまたは複数の範囲のワークロードに適用されるポリシーを作成するためのインベントリフィルタの使用を示しています。

3 つの範囲が含まれている次の例を考えてみましょう。

- **Apps**

  およびその子範囲

  - **Apps:HR** および
  - **Apps:Commerce**

さらに、インベントリフィルタ PRODUCTION および NON-PRODUCTION は、それぞれ実稼働ホストと非実稼働ホストを指定します（単一範囲内または複数範囲内のホストに適用するインベントリフィルタを定義できます）。

次のポリシーが **Apps** 範囲で定義されていると仮定します。

```
DENY PRODUCTION -> NON-PRODUCTION on TCP port 8000 (Absolute)
```

このポリシーは、**Apps** 範囲下のプライマリワークスペースで定義された絶対ポリシーであるため、**Apps** 範囲のメンバーであるすべての実稼働/非実稼働ホストに影響します。これには、子孫の範囲のメンバー（**Apps:HR** および **Apps:Commerce** の範囲に属するホスト）も含まれます。

ここで、**Apps:HR** 範囲に関連付けられているワークスペースでまったく同じポリシーが定義されている場合を考えてみましょう。このシナリオでは、ポリシーは**Apps:HR**範囲のメンバーである実稼働/非実稼働ホストにのみ影響します。より正確には、このポリシーにより、非実稼働 HR ホスト（存在する場合）のインバウンドルールは、**任意の**実稼働ホストからの TCP ポート 8000 での接続を拒否し、実稼働 HR ホスト（存在する場合）のアウトバウンドルールは、**任意の**非実稼働ホストへの接続要求をドロップします。

ポリシーの作成と検出

ポリシーの作成のベストプラクティス

- セグメンテーションプロセス全体の概要については、セグメンテーションとマイクロセグメンテーションを使用する前に （3 ページ）およびサブトピックを参照してください。
- ネットワーク全体に広く適用されるポリシーを手動で作成します。

  たとえば、ネットワークの外部からワークロードへの不要なトラフィックをブロックしたり、脆弱なホストを検疫したりします。

  - 範囲ツリーの最上位またはその近くにある範囲で手動ポリシーを作成します。

    たとえば、ネットワークの外部からネットワーク内のすべてのホストへのすべてのトラフィックをブロックするには、ツリーの最上位の範囲内にポリシーを配置します。

  - 一部のワークロードの一般的なポリシーを上書きできるようにする場合（たとえば、上記の例に従ってネットワークの外部からの一般的なアクセスをブロックし、一部のワークロードにはネットワークの外部からアクセスできるようにする場合）、高レベルのポリシーをデフォルトポリシーとして作成します。次に、該当するワークロードの固有のポリシーを作成します。

  - ポリシーの作成を迅速化するために、テンプレートの使用を検討してください。

  - ポリシーの手動作成 （525 ページ）、特定の目的のためのポリシー （527 ページ）、およびポリシーテンプレート （529 ページ）を参照してください。

- （オプション）最初に、ツリーの最上位近くの範囲でポリシーを自動的に検出し、ツリーのブランチ内のすべての範囲について、既存のすべてのトラフィックを許可し、将来の不要なトラフィックを制限できる粗いポリシーを作成します。その後、不要なトラフィックからネットワークをより適切に保護するきめ細かいポリシーを作成することができます。

  詳細については、1 つの範囲または範囲ツリーのブランチのポリシーの検出 （538 ページ）とポリシーの自動検出 （534 ページ）を参照してください。

- よりきめ細かいポリシーを検出する準備ができたら、範囲ツリーの最下位またはその近くにある範囲（特に個々のアプリケーションの範囲）のポリシーを自動的に検出します。

  詳細については、1つの範囲または範囲ツリーのブランチのポリシーの検出（538ページ）とポリシーの自動検出（534ページ）を参照してください。

- フェールオーバー、バックアップからの復元、1年に1回のアクティビティなど、一般的ではないまたは頻度の低いアクティビティとシナリオに対応するポリシーがあることを確認してください。

- アプリケーションに必要なトラフィックを特定して許可した後に、発生してはならないトラフィックを探して、そのようなインスタンスをブロックします。

  最初に、最も機密性の高いアプリケーションとの間のトラフィックを確認します。

  たとえば、顧客向けのWebアプリケーションから最高機密の研究開発アプリケーションのデータベースへのトラフィックを確認した場合は、調査します。

- 同僚と協力して、正しいポリシーが正しいワークロードに適用されていることを確認します。

- 最初に、ポリシーを適用する場合は、キャッチオールを[許可（Allow）]に設定することを検討してください。次に、トラフィックをモニターして、キャッチオールルールに一致するものを確認します。キャッチオールルールに一致する必要なトラフィックがない場合は、キャッチオールを[拒否（Deny）]に設定できます。

ポリシーの手動作成

通常は、ネットワーク全体に広く適用されるポリシーを手動で作成します。

たとえば、次の目的のためにポリシーを手動で作成できます。

- すべての内部ワークロードから、NTP、DNS、Active Directory、または脆弱性スキャンサーバーへのアクセスを許可する。
- 明示的に許可されていない限り、組織外のすべてのホストからネットワーク内のホストへのアクセスを拒否する。
- 脆弱なワークロードを検疫する。

より詳細に適用されるポリシーでオーバーライドできない絶対ポリシーと、より具体的なポリシーが存在する場合にオーバーライドできるデフォルトポリシーを作成できます。

通常、手動ポリシーは、ツリーの最上位に近い範囲に対して作成します。

始める前に

- （オプション）[防御（Defend）]>[ポリシーテンプレート（Policy Templates）]から使用可能なテンプレートのいずれかを使用することを検討してください。

- （オプション）同じポリシーを受信する必要があるワークロードのセットがあることがわかっている場合は、インベントリフィルタを使用してそれらをグループ化し、そのセットにポリシーを簡単に適用できるようにします。インベントリフィルタは、1 つの範囲にのみ適用することも、任意の範囲内のワークロードに適用することもできます。インベントリフィルタの作成 （453 ページ）を参照してください。
- この範囲内のワークロードがこの範囲内にあることが期待されているワークロードであることを確認してください。範囲内のワークロードの表示 （516 ページ）を参照してください。

手順

ステップ 1 [防御（Defend）] > [セグメンテーション（Segmentation）] をクリックします。

ステップ 2 左側のリストで、ポリシーを作成する範囲を検索するか、その範囲に移動します。

ステップ 3 ポリシーを作成する範囲とワークスペースをクリックします。

この範囲のワークスペースをまだ作成していない場合は、ワークスペースの作成 （515 ページ）を参照してください。

ステップ 4 [ポリシーの管理（Manage Policies）] をクリックします。

ステップ 5 まだ選択されていない場合は、[ポリシー（Policies）] タブをクリックします。

ステップ 6 [ポリシーを追加（Add Policy）] をクリックします。

[ポリシーの追加（Add Policy）] ボタンが表示されない場合は、[ポリシーの追加（Add Policy）] ボタンが使用できない場合 （527 ページ）を参照してください。

ステップ 7 情報を入力します。

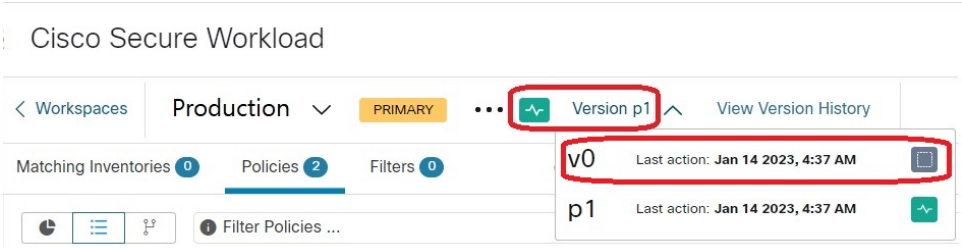
- [絶対（Absolute）] チェックボックスの詳細については、ポリシーのランク：絶対、デフォルト、キャッチオール （522 ページ）を参照してください。通常は、例外を想定しないポリシーを作成する場合に、このチェックボックスを有効にします。
- [優先順位（Priority）] は、リスト内のポリシーの順序を設定します。ポリシーの順序設定の詳細については、ポリシーの優先順位 （608 ページ）とサブトピックを参照してください（ポリシーの順序は後から設定できます）。
- コンシューマとプロバイダーを範囲全体にすることができます。または、インベントリフィルタ（または、最適ではありませんが、同じワークスペース内のクラスタ）を使用してワークロードのグループを作成している場合は、それらを選択できます。

次のタスク

[キャッチオール（Catch-all）] アクションがワークスペースに適していることを確認します。ポリシーのランク：絶対、デフォルト、キャッチオール （522 ページ）を参照してください。

[ポリシーの追加（Add Policy）] ボタンが使用できない場合

ポリシーを作成しようとしているときに [ポリシーの追加（Add Policy）] ボタンが使用できない場合は、ページの上部に表示されているバージョンをクリックし、灰色の正方形で示される最新の "v" バージョンを選択します。



特定の目的のためのポリシー

ネットワーク外部からのトラフィックをブロックするための InfoSec ポリシーの作成

ネットワークの外部からネットワークに入るトラフィックを制御する完全なポリシーセットをすばやく作成するには、この手順を使用します。デフォルトのポリシーセットは、共通のポートとプロトコルを使用するトラフィックのみを許可し、他のすべてのトラフィックを拒否します。必要に応じて、デフォルトのポリシーセットを変更できます。

始める前に

次の条件が満たされている場合は、この手順を使用します。

- 範囲ツリーのルート範囲のすぐ下に、**Internal** という名前の範囲がある。

  この範囲のメンバーに、内部ネットワーク上のすべてのワークロードを含むサブネットが含まれている、または含まれる予定である。

- Internal 範囲でまだポリシーが定義されていない。

（注） または、[防御（Defend）]>[ポリシーテンプレート（Policy Templates）] から入手可能な **InfoSec** テンプレートを使用し、いくつかの追加手順を行ってこれを実現することもできます。

手順

ステップ 1 [防御（Defend）]>[セグメンテーション（Segmentation）] を選択します。

ステップ 2 **Internal** 範囲をクリックし、プライマリワークスペースをクリックします。

プライマリワークスペースがまだ存在しない場合は、[+] ボタンをクリックして作成します。

ステップ 3 [ポリシーの管理（Manage Policies）] をクリックします。

ステップ 4 [InfoSecポリシーの追加（Add InfoSec Policies）] をクリックします。

ステップ 5 プロトコルとポートを含む、リスト内のすべてのポリシーが必要なポリシーであることを確認し、必要に応じてポリシーを削除および変更します。

ステップ 6 [作成（Create）] をクリックします。

次のタスク

（オプション）特定の外部トラフィックを特定のワークロードに許可するポリシーなど、追加のポリシーを Internal 範囲に追加します。

具体的なポリシーは、リスト内でより一般的なポリシーの下に配置します。

差し迫った脅威に対処するポリシーの作成

差し迫った脅威に対処する必要がある場合は、対象を絞り込んだ絶対ポリシーを範囲ツリーの最上位またはその近くの範囲に手動で追加してから、その範囲のプライマリワークスペースを適用することができます。

脅威を修復した後は、そのポリシーを削除し、ワークスペースを再適用することができます。

脆弱なワークロードを検疫するポリシーの作成

次の操作を実行できます。

- 事前にポリシーを作成して、特定の既知の脆弱性または指定した脆弱性の重大度しきい値を持つワークロードを自動的に検疫します。
- オンザフライでポリシーを作成して、十分に問題があると判断した既知の脆弱性が検出されたワークロードをすぐに検疫します。

このトピックでは、このいずれかを行うプロセスの概要を示します。

始める前に

必要になる可能性があるポリシーについては、脆弱性ダッシュボード（969ページ）を参照してください。

手順

ステップ 1 検疫する脆弱性または脆弱性の重大度しきい値を定義するインベントリフィルタを作成します。

a) ウィンドウの左側にあるナビゲーションバーから、[整理（Organize）]>[インベントリフィルタ（Inventory Filters）] を選択します。

b) [インベントリフィルタの作成（Create Inventory Filter）] をクリックします。

c) [クエリ（Query）] の横にある [(i)] ボタンをクリックし、**CVE** と入力して、関連するフィルタオプションを表示します。
d) 検疫するワークロードを決定するフィルタ条件を入力します。
e) [クエリを所有権の範囲に制限する（Restrict query to ownership scope）] が選択されていないことを確認してください。

ステップ 2 影響を受けるワークロードを検疫するポリシーを作成します。

全般的な手順については、ポリシーの手動作成 （525 ページ）を参照してください。

推奨事項：

- **Internal** 範囲内、または範囲ツリーの上位付近にあるその他の範囲内にポリシーを作成します。
- 例外を許可する場合を除き、ポリシーは絶対ポリシーである必要があります。例外に対処するには、ポリシーを作成してください。
- コンシューマとプロバイダーに個別のポリシーを作成します。
- 各ポリシーの優先順位を低い数値に設定して、リスト内の他のポリシーよりも前にヒットするようにします。
- アクションを [拒否（Deny）] に設定します。

ステップ 3 ポリシーを確認、分析、および適用します。

次のタスク

トラフィックがこのポリシーにヒットしたときに通知されるようにアラートを作成して、問題を修正し、脆弱なワークロードへのトラフィックを復元できるようにします。アラート （779 ページ） を参照してください。

ポリシーテンプレート

ポリシーテンプレートを使用して、同様のポリシーセットを複数のワークスペースに適用できます。

Cisco Secure Workload には、いくつかの事前定義されたテンプレートが含まれています。また、独自のテンプレートを作成することもできます。

ポリシーテンプレートには、ルート範囲の範囲所有者機能が必要です。

システム定義ポリシーテンプレート

使用可能なポリシーテンプレートを表示するには、[防御（Defend）]>[ポリシーテンプレート（Policy Templates）] を選択します。

ポリシーテンプレートを使用するには、テンプレートの適用 （533 ページ）を参照してください。

システム定義テンプレートを変更するには、JSON ファイルをダウンロードして編集し、アップロードします。

カスタム ポリシー テンプレートの作成

ポリシーテンプレートの JSON スキーマ

ポリシーテンプレート JSON スキーマは、ワークスペースのエクスポートのスキーマを模倣するように設計されています。ワークスペースで一連のポリシーを作成し、それを JSON としてエクスポートし、JSON を変更してから、ポリシーテンプレートとしてインポートできます。

属性	タイプ	説明
name	string	（オプション）インポート時にテンプレートの名前として使用されます。
説明	string	（オプション）適用プロセス中に表示されるテンプレートの説明。
パラメータ	パラメータオブジェクト	テンプレートパラメータ（下記を参照）。
absolute_policies	ポリシーオブジェクトの配列	（オプション）絶対ポリシーの配列。
default_policies	ポリシーオブジェクトの配列	（必須）デフォルトポリシーの配列、空にすることが可能。

パラメータオブジェクト

パラメータオブジェクトはオプションですが、テンプレートのパラメータとしてフィルタを動的に定義するために使用できます。パラメータは、`consumer_filter_ref` または `provider_filter_ref` ポリシー属性を使用して参照されます。

パラメータオブジェクトのキーは参照名です。値は、必須の `"type": "Filter"` とオプションの説明を含むオブジェクトです。パラメータオブジェクトの例を以下に示します。

```
{
  "parameters": {
    "HTTP Consumer": {
      "type": "Filter",
      "description": "Consumer of the HTTP and HTTPS service"
    },
    "HTTP Provider": {
      "type": "Filter",
      "description": "Provider of the HTTP and HTTPS service"
    }
```

```
    }
  }
```

パラメータは、ポリシーオブジェクトで参照できます（例：`"consumer_filter_ref": "HTTP Consumer"` または `"provider_filter_ref": "HTTP Provider"`）。

特殊なパラメータ参照

いくつかの特殊な参照は、自動的にフィルタにマッピングされます。パラメータとして定義する必要はありません。

Ref	説明
_workspaceScope	テンプレートが適用されているワークスペースの範囲に解決されます。
_rootScope	ルート/トップレベルの範囲に解決されます。

ポリシー オブジェクト

ワークスペースエクスポート JSON との互換性を維持するため、ポリシーオブジェクトには、コンシューマとプロバイダーの複数のキーが含まれています。それらは次のように解決されます。

```
if *_filter_ref is defined
  use the filter resolved by that parameter
else if *_filter_id is defined
  use the filter referenced by that id
else if *_filter_name is defined
  use the filter that has that name
else
  use the workspace scope.
```

上で定義したようにフィルタを解決できない場合、適用時とアップロード時の両方でエラーが返されます。

属性	タイプ	説明
action	string	（オプション）ポリシーのアクション、ALLOW または DENY（デフォルトは ALLOW）。
priority	integer	（オプション）ポリシーの優先順位（デフォルトは100）。
consumer_filter_ref	string	パラメータへの参照。
consumer_filter_name	string	名前によるフィルタへの参照。

属性	タイプ	説明
consumer_filter_id	string	定義された範囲またはインベントリフィルタの ID。
provider_filter_ref	string	パラメータへの参照。
provider_filter_name	string	名前によるフィルタへの参照。
provider_filter_id	string	定義された範囲またはインベントリフィルタの ID。
l4_params	l4params の配列	許可されたポートとプロトコルのリスト。
属性	タイプ	説明
proto	整数	プロトコル整数値（NULL はすべてのプロトコルを意味します）。
port	integer	ポートの包含範囲。[80, 80] または [5000, 6000] など（NULL はすべてのポートを意味します）。

L4param オブジェクト

属性	タイプ	説明
proto	整数	プロトコル整数値（NULL はすべてのプロトコルを意味します）。
port	integer	ポートの包含範囲。[80, 80] または [5000, 6000] など（NULL はすべてのポートを意味します）。

テンプレートのサンプル

```
{
  "name": "Allow HTTP/HTTPS and SSH",
  "parameters": {
    "HTTP Consumer": {
      "type": "Filter",
      "description": "Consumer of the HTTP and HTTPS service"
    },
    "HTTP Provider": {
```

```
        "type": "Filter",
        "description": "Provider of the HTTP and HTTPS service"
      }
    },
    "default_policies": [
      {
        "action": "ALLOW",
        "priority": 100,
        "consumer_filter_ref": "__rootScope",
        "provider_filter_ref": "__workspaceScope",
        "l4_params": [
          { "proto": 6, "port": [22, 22] },
        ]
      },
      {
        "action": "ALLOW",
        "priority": 100,
        "consumer_filter_ref": "HTTP Consumer",
        "provider_filter_ref": "HTTP Provider",
        "l4_params": [
          { "proto": 6, "port": [80, 80] },
          { "proto": 6, "port": [443, 443] }
        ]
      }
    ]
}
```

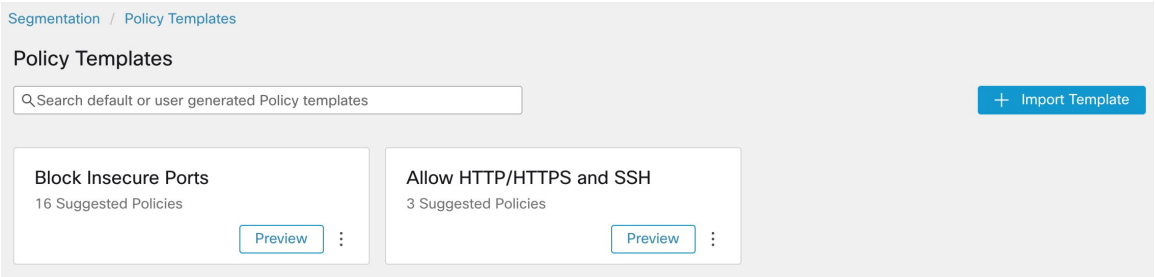
テンプレートインポート

ポリシーテンプレートは、メインの [セグメンテーション（Segmentation）] ページからアクセスできる [ポリシーテンプレート（Policy Templates）] ページに表示されます。この場所で [テンプレートのインポート（Import Template）] ボタンを使用してテンプレートをインポート/アップロードできます。

テンプレートは、アップロード時に正確性が検証されます。問題をデバッグするのに役立つエラーのリストが表示されます。

テンプレートがアップロードされると、テンプレートの適用、ダウンロード、または名前と説明の更新を行うことができます。

図 *265 :* 利用可能なテンプレートの表示



テンプレートの適用

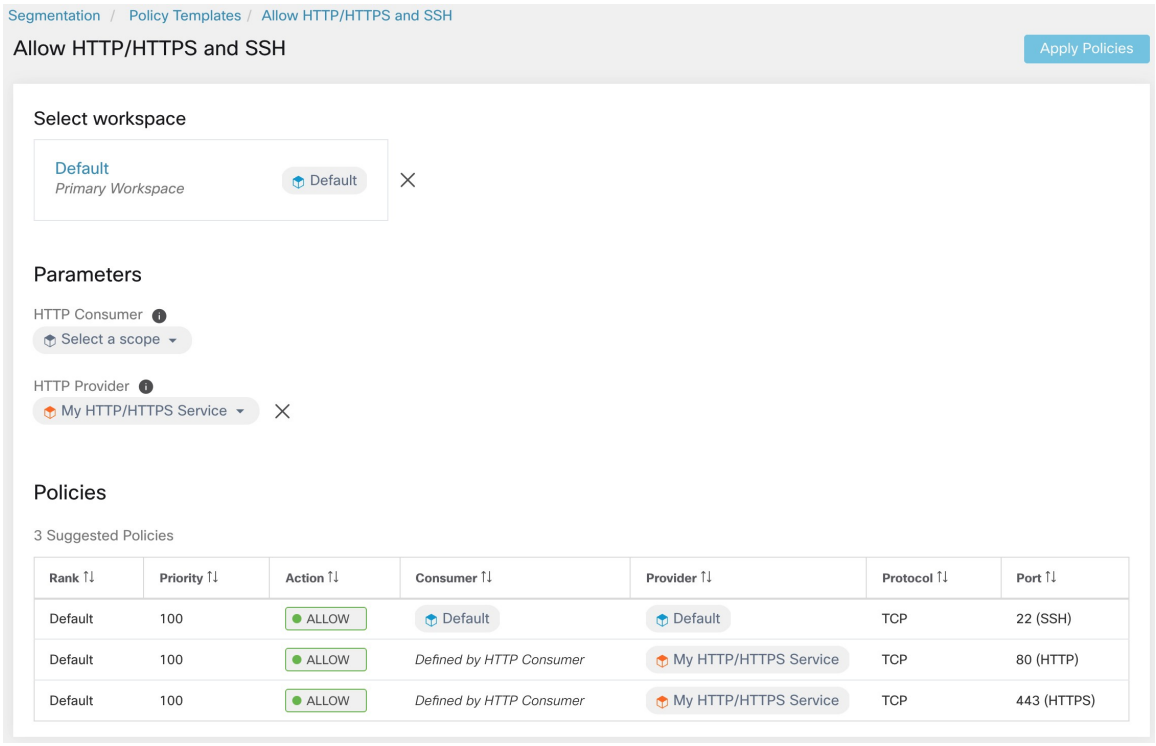
テンプレートをワークスペースに適用するには、いくつかの手順を実行します。

1. プレビューするテンプレートを選択します。

2. テンプレートを適用するワークスペースを選択します。
3. 必要に応じてパラメータを入力します。
4. ポリシーを確認します。
5. ポリシーを適用します。

ポリシーは、選択したワークスペースの最新バージョンに追加されます。テンプレートを介して作成されたポリシーは、`From Template? = true` フィルタを使用してフィルタリングできます。

図 *266* : ポリシーテンプレートの適用



ポリシーの自動検出

自動ポリシー検出は、ポリシー検出とも呼ばれ、以前はアプリケーション依存関係マッピング（ADM）と呼ばれていました。自動ポリシー検出は既存のトラフィックフローとその他のデータを使用して、次の処理を実行します。

- 既存の正常に完了したネットワークアクティビティに基づいて、一連の「許可」ポリシーを提案します。

  これらのポリシーの目的は、組織が必要とするトラフィックを識別し、他のすべてのトラフィックをブロックすることです。

- コンピューティング動作の類似性に基づいて、ワークロードをクラスタにグループ化します。

  たとえば、アプリケーションに複数の Web サーバーが含まれている場合、それらは一緒にクラスタ化される可能性があります。

  詳細については、「クラスタ （596 ページ）」を参照してください。

各範囲のポリシーを検出できます。通常、範囲のポリシーは、範囲ツリーの最下位付近（アプリケーションレベルなど）に表示されます。ただし、初期展開では、より上位の範囲でポリシーを検出することで、より改善されたポリシーを作成しながら、一般的で一時的なポリシーを使用することもできます。

必要に応じて何度でもポリシーを検出し、追加情報に基づいて推奨されるポリシーを改善することができます。

提案されたポリシーとクラスタの変更や承認を手動で行うことができます。これにより、ポリシーとクラスタは引き継がれ、後続のポリシー検出が実行されても変更されません。

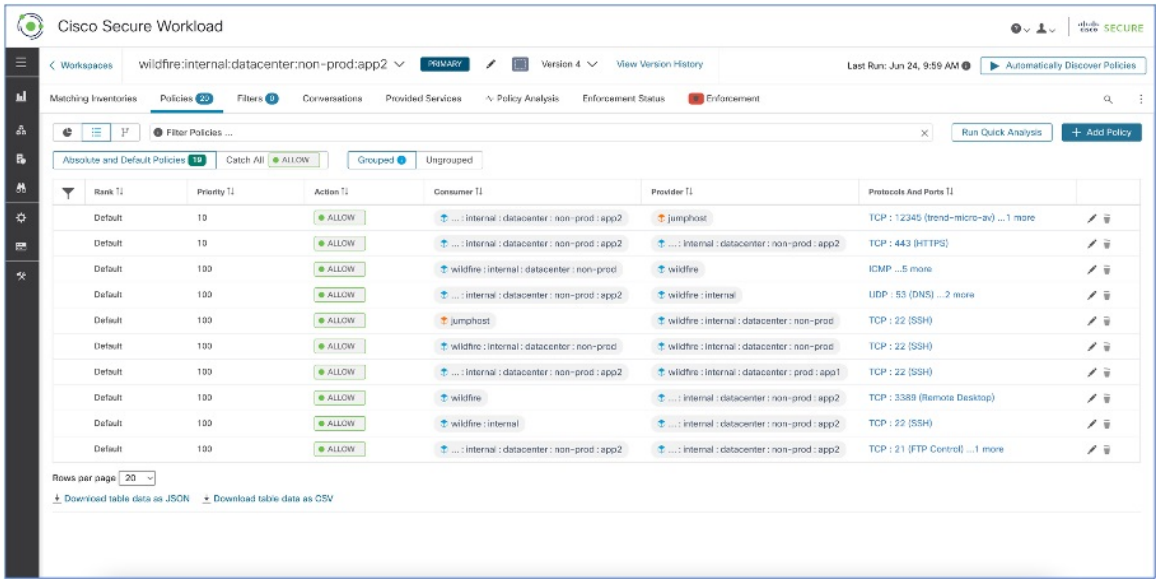
手動で作成したポリシーと検出されたポリシーの両方を、ワークスペースに含めることができます。

ポリシーを検出したら、適用する前に確認および分析します。

ポリシーの検出を開始するには、ポリシーを自動的に検出する方法 （536ページ）を参照してください。

詳細については、ポリシー検出の詳細 （535 ページ）を参照してください。

図 *267:* 例：自動検出されたポリシー



ポリシー検出の詳細

自動ポリシー検出に関する追加情報：

- 自動ポリシー検出では、選択した時間範囲内で少なくとも一方の端が範囲メンバーワークロードであるカンバセーションが考慮されます。範囲のメンバーシップは、最新の範囲定義のみに基づきます。以前のメンバーシップは考慮されません。
- デフォルトでは、ポリシー検出は通信フロー（「カンバセーション」）を分析してポリシーとクラスタを生成しますが、ワークロードで実行中のプロセスやロードバランサの設定といった他の情報をオプションで考慮に入れることができます。

  ポリシーの検出時にロードバランサとルータからのデータを含める （554ページ）を参照してください。
- 範囲内の任意のワークスペースでポリシーを検出できます。各ワークスペースの検出結果は、範囲内の他のワークスペースの結果とは無関係です。
- 自動ポリシー検出に関連する複雑な概念の詳細については、自動ポリシー検出の高度な機能 （545 ページ）およびポリシーの複雑さの対処 （607 ページ）を参照してください。

ポリシーを自動的に検出する方法

次のステップを実行します。いつでもポリシーを再検出できます。

必要に応じて同僚と協力して、これらの手順を完了します。

手順	操作手順	詳細情報
1	ワークロードインベントリをアップロードしてラベルを付け、ポリシー検出に情報を提供するためにフローデータを収集します。	「セグメンテーションとマイクロセグメンテーションを使用する前に （3 ページ）」およびサブトピックを参照してください。
2	次のどちらのポリシーを検出するかを選択します。 • 単一範囲内のワークロード • 範囲ツリーのブランチ内のすべての範囲内のワークロード	1つの範囲または範囲ツリーのブランチのポリシーの検出 （538 ページ）を参照してください。 （ポリシーはいつでも再検出できます）。
3	ポリシーを検出する範囲を選択します。	これは、単一範囲のポリシーを検出するか、範囲ツリーのブランチのポリシーを検出するかによって部分的に異なります。

手順	操作手順	詳細情報
4	ポリシーを検出するワークスペースを選択します。	ポリシーはプライマリワークスペースでのみ分析できるため、通常は範囲のプライマリワークスペースでポリシーを検出します（ただし、ワークスペースは後からいつでもプライマリに変更できます）。 選択した範囲にまだワークスペースがない場合は、ワークスペースの作成（515ページ）を参照してください。
5	ポリシー検出に含めるインベントリを確認します。	ポリシー検出が適用されるワークロードの確認（541 ページ）
6	（オプション）グループとして扱うワークロードをグループ化するインベントリフィルタを作成します。	インベントリフィルタの作成（453ページ）を参照してください。
7	ワークスペースの[キャッチオール（Catch-all）]アクションを設定します。	ポリシーのランク：絶対、デフォルト、キャッチオール（522 ページ）を参照してください
8	ポリシーを検出します。	ポリシーの自動検出（534 ページ） 「はじめる前に」の項の前提条件を満たしていることを確認してください。
9	ポリシー検出によって作成されるクラスタ（ワークロードのグループ）を表示および管理します。 （この手順は、単一範囲のポリシーを検出する場合にのみ適用されます。ツリーのブランチのポリシーを検出する場合、クラスタは生成されません）。	「クラスタ（596ページ）」およびサブトピックを参照してください。 提案されたクラスタを評価し、必要に応じてクラスタメンバーシップを編集して、永続化するクラスタを承認します（または、インベントリフィルタに変換する方がより適切です）。
10	ポリシーの継承やクロス範囲ポリシーなどの複雑さについて検討します。	ポリシーの複雑さの対処（607ページ）を参照してください。
11	生成されたポリシーを確認します。	自動検出されたポリシーの確認（633ページ）およびサブトピックを参照してください。
12	保持するポリシーを承認します。	ポリシーの承認（566 ページ）

手順	操作手順	詳細情報
13	必要に応じてポリシーを再検出し、追加のフローデータ、範囲メンバーシップの変更、またはその他の変更を反映します。	重要：自動ポリシー検出を再実行する前に（570ページ） ポリシー検出はいつでも再実行できます。 ポリシーを検出するたびに、ポリシーとクラスタを確認して承認します。
14	ライブ分析を実行して、ポリシーが実際のトラフィックにどのように影響するかを確認します。	ポリシーが期待どおりに機能していると確信できたら、ライブポリシー分析（641ページ）を開始します。 ポリシーを変更したり、ポリシーを再検出したりした場合は、ポリシー分析を再開します（現在のポリシーを分析するため）。
15	ポリシーを再検出したり、その他の変更を行ったりした場合は、ライブ分析を再開します。	ポリシーの変更後の、最新のポリシーの分析（651ページ）を参照してください。
16	ポリシーが重要なトラフィックをブロックしないことを確認できたら、ワークスペースを適用します。	「ポリシーの適用（654ページ）」およびサブトピックを参照してください。
17	適用が予想どおりに機能することを確認します。	適用が予想どおりに機能することの確認（665ページ）を参照してください。
18	（オプション）任意のワークスペースでポリシーを検出するときにオプションで適用されるデフォルトのポリシー検出設定を設定します。	デフォルトのポリシー検出設定（564ページ）およびリンク先のトピックを参照してください。 これらは詳細設定なので、変更する具体的な必要性がある場合にのみ変更することを推奨します。必要に応じて、プロセス中にいつでも変更できます。

1 つの範囲または範囲ツリーのブランチのポリシーの検出

特定の範囲のポリシーを検出するときにいずれかのオプションが使用できない場合は、選択が自動的に行われ、オプションの選択肢は表示されません。

表 *26 :* ポリシーの検出の対象

範囲ツリーのブランチ	単一範囲
Cisco Secure Workload の使用を開始する場合は、この方法を開始点として使用して、既存のトラフィックを許可すると同時に将来の脅威からネットワークを保護するための、一時的な大まかなポリシーのセットを迅速に生成します。	セグメンテーションポリシーを微調整し、許可されたすべてのフローが想定されるようにするために、この方法を使用します。ポリシーの数が少ないほど、調査が必要な既存の異常を簡単に確認できるようになります。
通常、この方法は範囲ツリーの最上位に近い範囲に使用します。 ブランチの最上位は、ツリー内の任意の範囲にすることができます。	通常、この方法は、単一のアプリケーション専用の範囲など、範囲ツリーの最下位またはその付近にある範囲に使用します。
1 つの範囲（選択したブランチの最上位にある範囲）でのみポリシーを検出します。	必要に応じて、ブランチ内の各範囲のポリシーを検出します。
選択した範囲内のすべてのワークロードと、すべての子および子孫の範囲が検出に含まれます。	子範囲のメンバーでもあるワークロードは、この範囲の検出には含まれません。 ポリシーは、[整理（Organize）]>[範囲とインベントリ（Scopes and Inventory）] ページでその範囲の [未分類のインベントリ（Uncategorized Inventory）] タブに表示されるワークロードに対してのみ生成されます。 子および子孫の範囲内のワークロードのポリシーを個別に検出できます。
ブランチ内の全範囲内のワークロードのポリシーすべてが、ブランチの最上位にある範囲内に存在します。	子および子孫の範囲内のワークロードのポリシーも作成すると仮定すると、ポリシーは複数の範囲内に存在します。
この方法では、通常、非常に多数のポリシーが生成されます。	この方法では、個々の範囲内に生成されるポリシーは少なくなります。
検出されたポリシーは、範囲全体に適用されます。このオプションでは、範囲内のワークロードのサブセットに固有のポリシーを作成できません。	このオプションでは、コンシューマやプロバイダーの範囲内のワークロードのサブセットに適用されるポリシーを生成できます（ワークロードは、生成されたクラスタや設定されたインベントリフィルタによってグループ化でき、ポリシーはこれらのサブセットにのみ適用されます）。

範囲ツリーのブランチ	単一範囲
すべてのポリシーがブランチの最上位にある単一範囲内に作成されるため、ポリシーのコンシューマとプロバイダーが異なる範囲にある場合、追加の手順は必要ありません。	異なる範囲内のコンシューマとプロバイダーの間のトラフィックを許可するには、追加の手順が必要です。 コンシューマとプロバイダーが異なる範囲にある場合：ポリシーオプション （615ページ）を参照してください。
子孫の範囲にフローデータを収集するエージェント、外部オーケストレータ、またはコネクタがある限り、インストールされたエージェントを持つメンバーワークロードが範囲にない場合でも、検出を実行できます。	範囲には、フローデータを収集するインストールされたエージェント、外部オーケストレータ、またはコネクタを持つメンバーワークロードが必要です。
このオプションは、ルート範囲の所有者とサイト管理者が使用できます。	この範囲のポリシーを作成する権限が必要です。
エージェントとカンバセーションの最大数は、オプションごとに異なります。ポリシーに関連する制限 （1298 ページ）を参照してください。	
このオプションは、以前は自動ポリシー検出用のディープポリシー生成の詳細設定オプションでした。動作は、変更されていません。	これは、以前は自動ポリシー検出用のデフォルトの動作でした。
詳細については、範囲ツリーのブランチのポリシーの検出：追加情報 （540ページ）を参照してください。	--

範囲ツリーのブランチのポリシーの検出：追加情報

- ポリシー検出が実行される範囲のメンバーであるかどうかに関係なく、対話のエンドポイントであるすべてのワークロードには、外部依存関係リストで指定された上から順序に、最も一致する範囲ラベルが割り当てられます。
- 範囲ツリーのブランチのポリシーを生成するときに使用できる詳細設定オプションについては、次を参照してください。
  - 冗長ポリシー削除の有効化 （561 ページ）
  - ポリシー圧縮 （557 ページ）および関連するサブトピック、階層型ポリシーの圧縮 （557 ページ）
- 現在、自動ポリシー検出で表示されるワークロードの数には、サブ範囲のメンバーではないワークロードのみが含まれます。

ポリシー検出が適用されるワークロードの確認

ポリシーを自動的に検出する前に、ポリシー検出のベースとなるワークロードが、実際に想定した一連のワークロードであることを確認します。検出されたポリシーは、これらのワークロードでエージェントによってキャプチャされたフローデータから生成されます。

始める前に

1 つの範囲または範囲ツリーのブランチのポリシーの検出 （538 ページ）のどのオプションを使用するかを決定します。

手順

ステップ 1 左側のナビゲーションメニューから、[防御（Defend）]>[セグメンテーション（Segmentation）] を選択します。

ステップ 2 ポリシーを検出する範囲をクリックします。

ステップ 3 ポリシーを検出するワークスペースをクリックします。

ステップ 4 [ポリシーの管理（Manage Policies）] をクリックします。

ステップ 5 [一致するインベントリ（Matching Inventories）] をクリックします。

ステップ 6 単一範囲のポリシーを検出する場合は、次の手順を実行します。

a) [未分類のインベントリ（Uncategorized Inventory）] をクリックします。

このページには、子範囲のメンバーではないワークロードが表示されます（標準の自動ポリシー検出では、子範囲のメンバーではないワークロードに対してのみ、この範囲内にポリシーとクラスタが生成されます）。

b) [IPアドレス（IP addresses）] をクリックします。

このページの IP アドレスには、Cisco Secure Workload エージェントがインストールされていません。

エージェントがインストールされていないため、次の場合を除き、この範囲の自動ポリシー検出中にこれらの IP アドレスは考慮されません。

- ポリシーがクラウドコネクタを介して管理されている。
- IP アドレスがコンテナベースのインベントリである。この場合、個々のワークロードが [ポッド（Pods）] タブに表示されます。
- ポリシー検出中に考慮されるこの範囲内のワークロードと通信するためにワークロードが発生する。

ポリシーを検出する前に、エージェントを必要とするワークロードにエージェントをインストールし、フローデータが蓄積されるまでしばらく待つことを検討してください。

c) [ワークロード（Workloads）] をクリックします。

ポリシーとクラスタは、このページのワークロードと、[IPアドレス（IP addresses）] タブの IP アドレスのうち、考慮するために上記で指定されている基準を満たす IP アドレスに対してのみ生成されます。

d) Kubernetes または OpenShift インベントリがある場合は、[サービス（Services）] タブと [ポッド（Pods）] タブも表示されます。

Kubernetes/OpenShift ワークロードにエージェントをインストールしている場合は、それらのタブのインベントリも確認します。

e) ロードバランサインベントリがある場合、そのインベントリは [サービス（Services）] タブに表示されます。

ステップ 7 ツリーのブランチのポリシーを検出する場合は、次の手順を実行します。

a) [すべてのインベントリ（All Inventory）] をクリックします。

このプロセスは、子範囲のメンバーでもあるかどうかに関係なく、この範囲内のすべてのワークロードのポリシーを生成します（ただし、クラスタは生成しません）。

b) [IPアドレス（IP addresses）] をクリックします。

このページの IP アドレスには、Cisco Secure Workload エージェントがインストールされていません。

エージェントがインストールされていないため、次の場合を除き、この範囲の自動ポリシー検出中にこれらの IP アドレスは考慮されません。

- ポリシーがクラウドコネクタを介して管理されている。
- IP アドレスがコンテナベースのインベントリである。この場合、個々のワークロードが [ポッド（Pods）] タブに表示されます。
- ポリシー検出中に考慮されるこの範囲内のワークロードと通信するためにワークロードが発生する。

ポリシーを検出する前に、これらのワークロードにエージェントをインストールし、フローデータが蓄積されるまでしばらく待つことを検討してください。

c) [ワークロード（Workloads）] をクリックします。

ポリシーは、このページのワークロードと、[IPアドレス（IP addresses）] タブの IP アドレスのうち、考慮するために上記で指定されている基準を満たす IP アドレスに対してのみ生成されます。

d) Kubernetes または OpenShift インベントリがある場合は、[サービス（Services）] タブと [ポッド（Pods）] タブも表示されます。

Kubernetes/OpenShift ワークロードにエージェントをインストールしている場合は、それらのタブのインベントリも確認します。

e) ロードバランサインベントリがある場合、そのインベントリは [サービス（Services）] タブに表示されます。

ステップ 8 ワークロードが想定どおりのセットであることを確認します。

ポリシーの自動検出

この手順を使用して、ネットワーク上の既存のトラフィックに基づいて推奨される許可ポリシーを生成します。

ポリシーはいつでも再検出できます。

始める前に

- ポリシーを効果的に自動検出するには、フローデータを収集する必要があります。

  これは通常、範囲内のワークロードにエージェントをインストール済みであるか、あるいはクラウドコネクタまたは外部オーケストレータを使用してデータを設定および収集済みであるかを意味します。

  自動ポリシー検出で使用されるフローサマリーデータは、現在6時間ごとに計算されています。したがって、Secure Workloadの初回の展開では、そのようなデータが利用可能になるまで、自動ポリシー検出はできません。

  一般に、フローデータが多いほど、より正確な結果が得られます。

  定期的に（月次、四半期、年次など）のみ発生するトラフィックを含めるために、ポリシーを適用する前に十分なデータを収集する必要があります。たとえば、アプリケーションが他の時間にはアクセスしないソースから情報を収集する四半期レポートを生成する場合、フローデータにそのレポート生成プロセスのインスタンスが少なくとも1つ含まれていることを確認してください。

- ポリシーを自動的に検出する方法 （536 ページ）のここまでの手順を完了します。
- ポリシー検出関連のポリシーに関連する制限 （1298 ページ）を満たします。

  必要に応じて、大きな範囲を小さな子範囲に分割します。

- ポリシーを検出する前に範囲の変更をコミットしないと、設定された除外フィルタが期待どおりにフローと一致（除外）しない可能性があります。変更の確定 （437 ページ）を参照してください。



重要 ポリシー検出を再実行する場合は、最初に重要な考慮事項を参照してください（重要：自動ポリシー検出を再実行する前に （570 ページ））。

手順

ステップ 1 **[防御（Defend）]** > **[セグメンテーション（Segmentation）]** の順に選択します。

ステップ 2 左側のペインの範囲ツリーまたは範囲のリストで、ポリシーを生成する範囲までスクロールするか、その範囲を検索します。

ステップ 3 範囲内のワークスペース（プライマリまたはセカンダリ）をクリックします。

ステップ 4 [ポリシーの管理（Manage Policies）] をクリックします。

ステップ 5 [ポリシーを自動的に検出（Automatically Discover Policies）] をクリックします。

ステップ 6 ブランチまたは範囲全体のポリシーを検出するオプションが表示された場合は、オプションを選択します。

オプションが表示されない場合は、ポリシーを検出している範囲で使用できるオプションは 1 つだけです。

詳細については、「1 つの範囲または範囲ツリーのブランチのポリシーの検出（538 ページ）」を参照してください。

ステップ 7 含めるフローデータの時間範囲を選択します。

何度か試して、適切な時間範囲を見つけてください。最適な結果を得るため、何度でもポリシーを生成できます。

時間範囲を短くすると、結果の生成が速くなり、少なくなる可能性があります。

一般に、時間範囲が長いほど、より正確なポリシーが生成されます。ただし、範囲の定義が変更されている場合は、変更が行われる前の日付を含めないでください。

該当する場合は、時間範囲には定期的にのみ発生するトラフィック（月次、四半期、年次など）を含める必要があります。たとえば、アプリケーションが他の時間にはアクセスしないソースから情報を収集する四半期レポートを生成する場合、時間範囲にはそのレポート生成プロセスのインスタンスが少なくとも 1 つ含まれていることを確認してください。

過去 30 日を超える時間範囲を設定するには、[カスタム（Custom）] 範囲を選択し、時間選択ウィジェットのドロップダウンの下に希望の開始時刻と終了時刻を入力します。

ステップ 8 （オプション）[詳細設定（Advanced Settings）] を指定します。

通常は、最初の検出の実行では [詳細設定（Advanced Settings）] は変更せず、特定の問題に対処するために必要な場合にのみ変更を行うことが推奨されます。

詳細については、自動ポリシー検出の詳細設定 （554 ページ）を参照してください。

ステップ 9 [ポリシーの検出（Discover Policies）] をクリックします。生成されたポリシーは、このページに表示されます。

次のタスク

- 進行中の自動ポリシー検出の停止 （545 ページ） を表示します。
- ポリシーを自動的に検出する方法（536 ページ）に戻り、表の次のステップに進みます。
- ポリシーはいつでも再検出できます。最初に実行する必要があるアクションについては、重要：自動ポリシー検出を再実行する前に （570 ページ）を参照してください。

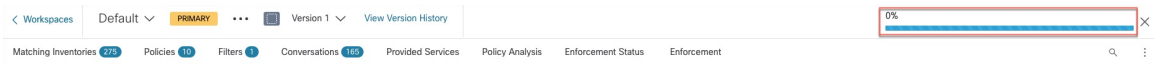
進行中の自動ポリシー検出の停止

自動ポリシー検出の進行状況は常にヘッダーに表示されます。他のワークスペースに移動しても、進行状況には影響しません。

進行中に実行を停止するには、[中止（abort）] ボタンをクリックします。

実行が完了すると、メッセージが表示されます。成功した場合、[クリックして結果を表示（Click to see results）] をクリックして、実行前後の変更を示す別のビューに移動します。自動ポリシー検出が失敗した場合は、別のメッセージが表示され、場合によっては理由が示されます。

図 *268 :* 自動ポリシー検出の進行状況



自動ポリシー検出の高度な機能

検出実行の時間範囲のみを指定する必要がある場合があります。必要に応じて、詳細オプションを設定できます。

各ワークスペースの詳細オプションを設定するか、すべてのワークスペース（ルート範囲全体）のデフォルトを設定してから、必要に応じて個々のワークスペースの設定を変更することができます。

表 *27 :* 自動ポリシー検出の詳細オプションの設定

単一のワークスペースの場合	すべてのワークスペースの場合
個々のワークスペース（1列目）のオプションの説明が、すべてのワークスペース（2列目）にも適用されます	
外部依存関係 （549 ページ）	デフォルトのポリシー検出設定 （564ページ）
自動ポリシー検出の詳細設定 （554 ページ）	デフォルトのポリシー検出設定 （564ページ）
除外フィルタ （545 ページ）	デフォルトの除外フィルタ （565 ページ）

除外フィルタ

特定のフローが不要なポリシーを生成している場合は、除外フィルタを使用してそのフローを自動ポリシー検出から除外することができます。

たとえば、最終的な許可リストモデルでICMPなどの特定のプロトコルを禁止するには、プロトコルフィールドをICMPに設定して除外フィルタを作成できます。

（注）

- 除外フィルタに一致するカンバセーションは、ポリシーの生成とクラスタリングの目的で除外されますが、赤色の [除外（Exclude）] アイコン付きでカンバセーションビューに残ります（「カンバセーション」のテーブルビューを参照）。同様に、そのようなカンバセーションでのワークスペースインシデントのワークロードも表示されたままになります。
- ワークスペースのクラスタ定義やフィルタ定義を使用する除外フィルタは、プライマリワークスペースでのみ有効です（それ以外の場合、そのクラスタ定義はラベルシステムに対する可視性はないため、一致するカンバセーションは除外されません）。
- 除外フィルタはバージョン管理されています。変更を追跡するには、「アクティビティログとバージョン履歴」を参照してください。
- 除外フィルタ数の制限については、「ポリシーに関連する制限（1298ページ）」を参照してください。

次のいずれか1つまたは両方を作成し、ポリシーの検出時にいずれかまたは両方を有効にできます。

- 各ワークスペースの除外フィルタのリスト。
- テナント内のすべてのワークスペースで使用できるデフォルトの除外フィルタのリスト。

デフォルトのポリシー検出設定のいずれか1つまたは両方のリストを有効または無効にすることもできます。

手順については、「除外フィルタの構成、編集、または削除（546ページ）」および「除外フィルタを有効または無効にする （549 ページ）」を参照してください。

除外フィルタの構成、編集、または削除

この手順を使用して、1 つのワークスペースの除外フィルタのリスト、またはすべてのワークスペースで使用可能な既定の除外フィルタのリストを作成できます。

手順

ステップ 1 次のいずれかを実行します。

目的	操作手順
特定のワークスペースの除外フィルタを設定する	ワークスペースに移動し、次のいずれかを実行します。 • [ポリシーの管理（Manage Policies）] をクリックし、次にページの右上付近にある ⋮ をクリックして、[除外フィルタ（Exclusion Filters）] を選択します。 • 自動ポリシー検出の設定ページで、[詳細設定（Advanced Configurations ）] セクションの [除外フィルタ（Exclusion filters）] リンクをクリックします。 • 検出されたポリシーを削除します。除外フィルタを作成するオプションが表示されます。
すべてのワークスペースで使用可能な既定の除外フィルタを設定する	1. **[防御（Defend）]** > **[セグメンテーション（Segmentation）]** の順に選択します。 2. ページの右側にあるキャレット記号をクリックして[ツール（Tools）] メニューを展開し、[デフォルトのポリシー検出設定（Default Policy Discovery Config）] を選択します。 3. ページの下部までスクロールします。 4. [デフォルトの除外フィルタ（Default Exclusion Filters）] をクリックします。

ステップ 2 除外フィルタを作成するには、[除外フィルタの追加（Add Exclusion Filter）] をクリックします。

ステップ 3 ポリシー検出時に考慮しないフローのパラメータを指定します。

すべてのフィールドに値を入力する必要はありません。空白のフィールドは、一致するフローのワイルドカードとして扱われます。

除外フィルタのすべてのフィールドに一致するカンバセーションは、ポリシーの作成とクラスタリングの目的においては無視されます。

オプション	説明
コンシューマ	コンシューマアドレスが、選択した範囲またはインベントリフィルタ（または、ワークスペース固有の除外フィルタの場合のみ、クラスタ）のメンバーである、カンバセーションに一致させます。新しいカスタムフィルタを作成することにより、任意のアドレス空間を指定できます。

オプション	説明
プロバイダ（Provider）	プロバイダーアドレスが、選択した範囲またはインベントリフィルタ（または、ワークスペース固有の除外フィルタの場合のみ、クラスタ）のメンバーである、カンバセーションに一致させます。新しいカスタムフィルタを作成することにより、任意のアドレス空間を指定できます。
[Protocol]	指定されたプロトコルとのカンバセーションに一致させます。
ポート（Port）	指定されたポートまたはポート範囲に一致するプロバイダー（サーバー）ポートとのカンバセーションに一致させます。ダッシュ区切りを使用してポート範囲を入力します（例：「100-200」）。

ステップ 4 除外フィルタを編集または削除するには、該当する行にカーソルを合わせて、[編集（Edit）] および [削除（Delete）] ボタンを表示します。

ステップ 5 デフォルトの除外フィルタを設定する場合は、次の手順を実行します。

設定したフィルタを使用する準備ができたら、[デフォルトのポリシー検出設定（Default Policy Discovery Config）] ページに戻り、[保存（Save）] をクリックして、個々のワークスペースで変更を使用できるようにします。

次のタスク

重要 除外フィルタは、設定されているワークスペースでデフォルトで有効になっています。

デフォルトの除外フィルタは、すべてのワークスペースでデフォルトで有効になっています。

デフォルトのポリシー検出設定では、両方のタイプの除外フィルタがデフォルトで有効になっています。

ポリシーを検出する前に、次を実行します。

- 除外フィルタとデフォルトの除外フィルタを、次の場所で有効または無効にします
  - 各ワークスペース
  - [デフォルトのポリシー検出設定（Default Policy Discovery Config）] ページ

  この説明については、除外フィルタを有効または無効にする （549ページ） を参照してください。
- 範囲の変更をコミットしないと、予定されるフローとフィルタが一致しない（そのため除外される）可能性があります。変更の確定 （437 ページ） を参照してください。

除外フィルタを有効または無効にする

各ワークスペースで除外フィルタを作成したり、すべてのワークスペースに適用できるデフォルトの除外フィルタのセットを作成したりすることができます。

デフォルトでは、両方のタイプの除外フィルタが有効になっています。

変更するには、次の手順に従います。

- 単一のワークスペースの除外フィルタを有効または無効にするには：

  ワークスペースで、[ポリシーの管理（Manage Policies）] をクリックしてから、[ポリシーの自動検出（Automatically Discover Policies）] をクリックし、次に [詳細設定（Advanced Configurations）] の順にクリックします。 このワークスペースの除外フィルタおよび/またはデフォルトの除外フィルタを有効にすることができます。

- [デフォルトのポリシー検出設定（Default Policy Discovery Config）] で除外フィルタを有効または無効にするには：

  [防御（Defend）]>[セグメンテーション（Segmentation）] を選択し、ページの右側にあるキャレット記号をクリックして [ツール（Tools）] メニューを展開します。次に、[デフォルトのポリシー検出設定（Default Policy Discovery Config）] を選択します。 [詳細設定（Advanced Configurations）] までスクロールするか、これをクリックします。除外フィルタおよび/またはデフォルトの除外フィルタを有効にすることができます。

外部依存関係

外部依存関係は、（上級）クロス範囲ポリシーの作成（616ページ）で説明されているプロセスを使用する場合にのみ関連します。

外部依存関係の設定は、ポリシーが検出された範囲以外の範囲のメンバーであるワークロードとの間の通信が関与する、自動検出されたポリシーに適用されます（つまり、「外部ワークロード」が関与する通信）。

ポリシーが存在する範囲のメンバーではないワークロードは、外部ワークロードです。このようなワークロードは、（ポリシーが存在する範囲のメンバーである）ターゲットワークロードとの会話先です。

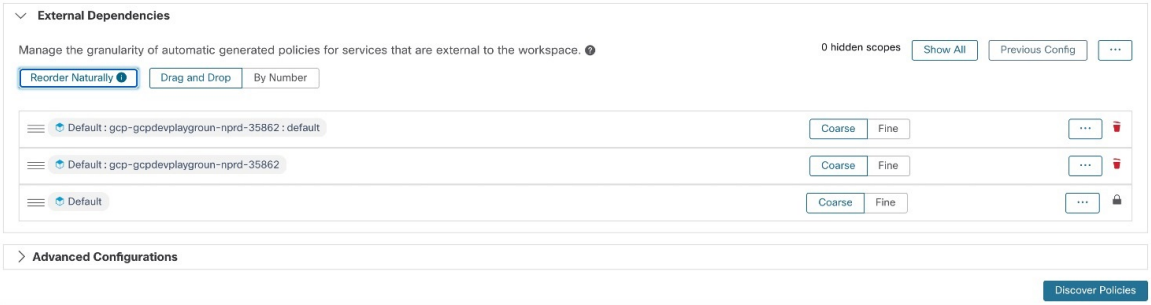
外部依存関係リストは、階層内のすべての範囲の順序付きリストです。リスト内の各範囲は、次のいずれかに設定されます。

- 特定のポリシーまたは（より安全な）改良ポリシーの生成、または
- より高度な範囲での大まかなポリシーの生成。こちらの方がより一般化しやすい場合があります（つまり、ポリシーの検出時に指定の時間範囲内で確認できなかった正当なフローを許可する可能性が高くなります）。

ポリシーの検出中に、ワークロードに一致する最初の範囲（またはクラスタ、またはインベントリフィルタ。以下を参照）を使用して「許可」ポリシーが生成されます。一致順（および結果的な粒度レベル）は、[外部依存関係（External Dependencies）] セクションに表示されるトップダウン方式のランク付けで決定されます。

デフォルトの範囲の順序が設定され、すべての範囲がデフォルトで[粗い（Coarse）]に設定されます。

図 *269*：デフォルトの外部依存関係



目的	操作手順
ワークスペースの外部依存関係を表示または微調整します。	ワークスペースに移動し、[ポリシーの自動検出（Automatically Discover Policies）]をクリックしてから、[外部依存関係（External Dependencies）]をクリックします。 範囲を並べ替え、それぞれの粒度オプションを選択するには、「ワークスペースの外部依存関係の微調整 （551 ページ）」を参照してください。
ルート範囲全体のデフォルトの外部依存関係を設定します。	デフォルトのポリシー検出設定 （564 ページ）を参照してください。

外部依存関係：範囲のサブセットを含むきめ細かいポリシー

オプションで、範囲間よりもきめ細かいレベルでポリシーを検出して、範囲内のワークロードの指定されたサブセットへのトラフィックを制御できます。

たとえば、アプリケーション内の特定のタイプのホスト（API サーバーなど）に固有のポリシーを作成する場合、これらのワークロードをアプリケーション範囲内のサブセットにグループ化できます。

範囲内のワークロードのサブセットに固有のポリシーを生成するには、ワークスペースの外部依存関係の微調整 （551 ページ）を参照してください。

外部依存関係の調査に関するヒント

次のヒントを使用して、ポリシーが存在するワークスペースに関連付けられた範囲のメンバーではないワークスペースを含むポリシーの自動ポリシー検出の動作を調べます。

ヒント

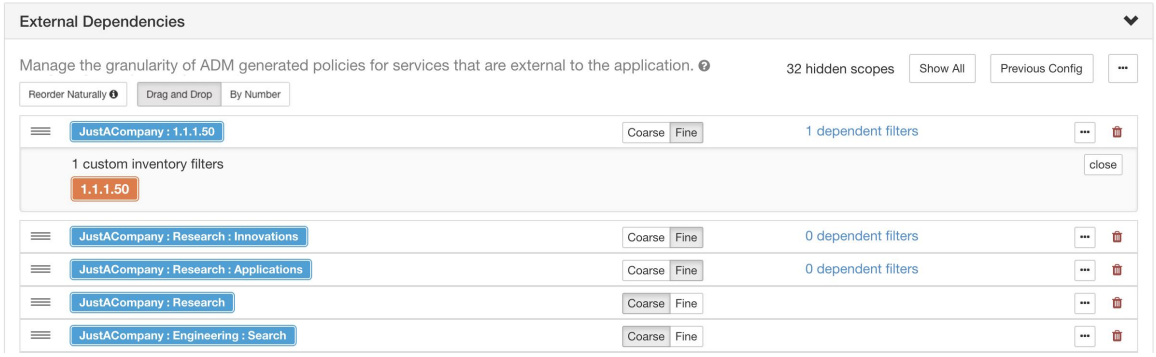
- リストを削除および再配置して、目的の粒度でポリシーを生成できます。たとえば、すべての Company:RTP サブ範囲を削除すると、Company:SJC 範囲に対してより高い粒度を維持しながら、個々のコンポーネントではなく Company:RTP 範囲全体に対する幅広いポリシーを生成できます。さらに、任意の範囲の横にある [詳細（Fine）] ボタンをクリックして、その範囲の下に定義されているより細かい候補の存在を確認できます。
- デフォルトでは、ルート範囲は外部依存関係リストの最下位のエントリとして構成されているため、自動ポリシー検出では可能な限り、常により具体的な範囲に対するポリシーが生成されます。最初は、比較的少数の大まかなポリシーを表示するために、ルート範囲を外部依存関係の最上位に一時的に配置できます。これで、自動ポリシー検出の後に、ワークスペースのすべての外部ポリシーが 1 つの範囲、つまりルート範囲にのみ接続されていることを確認できます（すべての外部ワークロードがルート範囲にマッピングされるため）。その結果、生成されるポリシーの数が少なくなり、調査と理解が容易になります。
- また、ワークスペースに関連付けられた範囲のメンバーであるすべてのワークロード（「内部ワークロード」）を 1 つのクラスタに一時的にバンドルし、クラスタを承認してから、ポリシーを検出することもできます。この場合も、クラスタリング（ワークスペースまたは範囲のサブパーティション化）が行われないため、一連のポリシーが減り、内部（内部ワークロードに接続）または外部（内部ワークロードを外部ワークロードに接続）するポリシーを表示できます。その後、内部ワークロードをバンドル解除したり、関心のある 1 つまたは複数の外部範囲をルートの上に配置したりすることで、徐々に詳細なポリシーを表示できます。
- **重要**ルート範囲を含むポリシーは、ネットワーク全体との間で送受信されるすべてのトラフィックを許可するため、常に注意深く調べてください。これは、ルート範囲が外部依存関係リストの下位に配置されていて、粗いポリシーの生成を意図していない場合に特に重要です。そのようなポリシーは、ワークスペース範囲に出入りするネットワーク全体のトラフィックに起因するポリシーではない可能性があります。むしろ、ルート範囲を超えて、より細かい範囲やインベントリフィルタの割り当てを受信できなかった複数の外部エンドポイントによってトリガーされる可能性があります。

  そのようなポリシーを監査するときは、関連する会話（「カンバセーション」を参照）を調べてエンドポイントを特定し、より細かい範囲またはインベントリフィルタに分類して、ルート範囲レベルでの安全性の低いポリシーを回避する必要があります。

ワークスペースの外部依存関係の微調整

ポリシーのプロバイダーがポリシーが検出されている範囲とは異なる範囲に属している場合、この手順を使用して、自動ポリシー検出時に（範囲全体ではなく）範囲内の指定されたワークロードのサブセット間でポリシーを作成します。

図 *270 :* 外部依存関係の微調整



始める前に

- 特定のポリシーを生成するワークロードのサブセットごとに、インベントリフィルタを設定します。任意の範囲で、任意の数のインベントリフィルタを作成できます。

  インベントリフィルタを作成するには、いくつかの方法があります。

  - 対象のクラスタをインベントリフィルタに変換する。

    （クラスタをインベントリフィルタに変換する （601 ページ）を参照）

    および/または

  - 新しいインベントリフィルタを作成する。

    インベントリフィルタの作成 （453 ページ）を参照してください。

  インベントリフィルタでは、次のオプションを有効にする必要があります。

  - [クエリを所有権の範囲に制限する（Restrict Query to Ownership Scope）]

    [範囲外のサービスを提供する（Provides a service external of its scope）]

- 外部依存関係の調査に関するヒント （550 ページ）も参照してください。

手順

ステップ 1 ポリシーを検出するワークスペースに移動します。

ステップ 2 [ポリシーを自動的に検出（Automatically Discover Policies）] をクリックします。

ステップ 3 [外部依存関係（External Dependencies）] をクリックします。

ステップ 4 必要に応じて、[すべての範囲を表示（Show All scopes）] をクリックします。

ステップ 5 （オプション）以前の設定を利用します。

- 最後にポリシーを検出したときにリストに加えた変更を再利用するには、[以前の設定（Previous Config）] をクリックします。

- グローバルな「デフォルトのポリシー検出設定」で外部依存関係を設定している場合は、[デフォルト設定（Default Config）] をクリックしてグローバルリストを使用できます。または、デフォルトのリストを取得した後、必要に応じて（そのワークスペースに対してのみ）変更し、[前の設定（revious Config）] を 1 回クリックすると、それ以降はカスタマイズされたバージョンを使用できます。

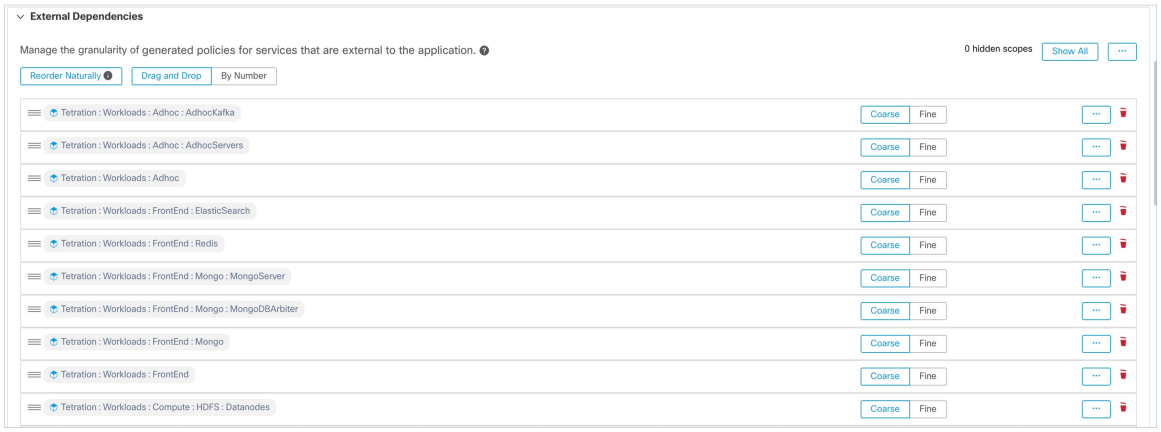
ステップ 6 必要に応じて範囲（および該当する場合はインベントリフィルタ）を並べ替えます。

ポリシーは、トラフィックに一致するリスト（上から順）の最初の範囲またはインベントリフィルタに基づいて適用されます。このためには、通常、トラフィックに一致する最も具体的なポリシーを適用する必要があるため、親（あまり具体的ではない）の上に子範囲（より具体的）が必要です。

- 新しい子範囲を最近作成した場合は、デフォルトでリストの一番下に追加されますが、リスト全体を並べ替えて、子範囲を親の上に配置します。

  （推奨）[自然に並べ替える（Reorder Naturally）] をクリックします。

  図 *271* : 自然に並べ替える



- （特定の理由がある場合）リストを手動で並べ替えるには、次の手順を実行します。
  - [ドラッグアンドドロップ（Drag and Drop）] をクリックします。
  - [番号順（By Number）] をクリックします。

    外部依存関係には、10 の倍数で優先順位の値が割り当てられます。値を変更して順序を変更します。

    番号を変更したら、[表示（View）] をクリックしてリストの順序を更新し、10 の倍数を優先順位のそれぞれに再割り当てします。

ステップ 7 各行の精度を指定します。

- 設定されたインベントリフィルタまたはクラスタに固有のポリシーを生成するには、各行の [細かい（Fine）] をクリックします。

  範囲全体に適用するポリシーを生成するには、[粗い（Coarse）] をクリックします。

- 範囲のすべてのサブ範囲に精度を適用するには、範囲の行の最後にある3つのドットのボタンをクリックします。

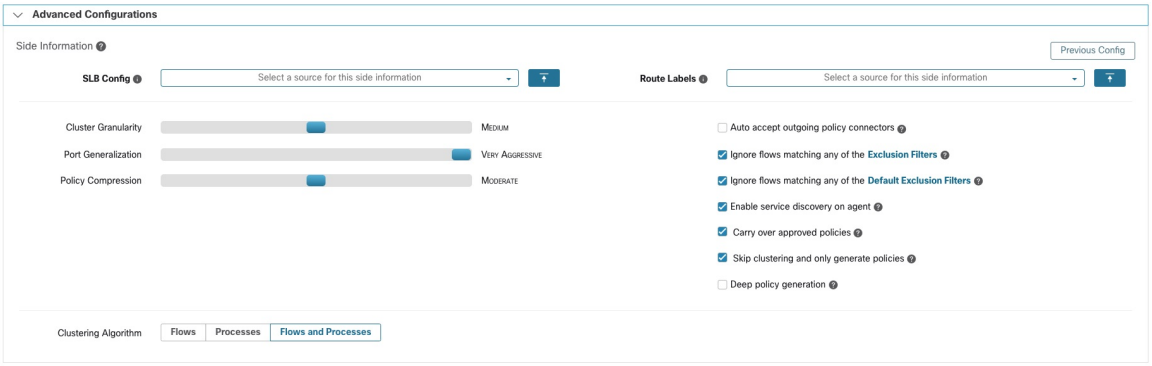
自動ポリシー検出の詳細設定

詳細設定を使用して、ポリシーを検出する際に追加情報を指定することや、特定の環境に適応させることができます。

通常、特別な理由がない限り、これらのオプションは変更しないことを推奨します。

- 特定のワークスペースのこれらの設定にアクセスするには、該当するワークスペースで[ポリシーの自動検出（Automatically Discover Policies）] をクリックします。
- すべてのワークスペースのデフォルトを変更するには、デフォルトのポリシー検出設定（564 ページ）を参照してください。

図 *272 :* 自動ポリシー検出の詳細設定



ポリシーの検出時にロードバランサとルータからのデータを含める

ロードバランサとルータからデータをアップロードして、自動ポリシー検出を通知できます。

次のオプションにアクセスするには、[自動ポリシー検出（Automatic Policy Discovery）] の設定で [詳細設定（Advanced Configurations）] をクリックし、[サイド情報（Side Informaton）] または [サイド情報（sideinfo）] セクションを確認します。

オプション	説明
SLB 構成 （ロードバランサ構成のアップロード）	正しい形式でロードバランサからデータをダウンロードするには、「高度なポリシー検出設定を実現するためのロードバランサ設定の取得」を参照してください。 ロードバランサ構成のアップロードでサポートされている形式： • **F5 BIG-IP** • **Citrix NetScaler** • **HAProxy** • その他： **正規化された JSON** スキーマを使用します。 サポートされていないロードバランサ構成はこのスキーマに変換する必要があります。 この単純なスキーマには、仮想 IP（VIP）とバックエンド IP に関する基本情報が含まれています。 サンプル JSON ファイルをダウンロードするには、[SLB構成（SLB Config）] の横にある情報ボタンをクリックします。
ルートラベルのアップロード	プロビジョニングされたサブネットまたはルートのリストをルータからアップロードして、事前にプロビジョニングされた一連のサブネットに基づいてホストを分割できます。自動ポリシー検出によって生成されるクラスタリングの結果は、アップロードされたデータで定義されているサブネットの境界にまたがることはありません。自動ポリシー検出が完了したら、結果を変更できます。 サンプルの JSON ファイルをダウンロードするには、[ルートラベル（Route Labels）] の横にある情報ボタンをクリックします。

（注） クラスタはパーティションの境界にまたがりません。つまり、自動ポリシー検出によって計算されたクラスタには、2つの異なるパーティションのターゲットワークロードは含まれません。パーティションは、アップロードされたロードバランサまたはルータのデータから計算されます。ただし、クラスタクエリ定義を変更すること（手動クラスタ編集）により、ワークロードをクラスタ間で自由に移動させたり、サイド情報のアップロードを無効化したりできます。

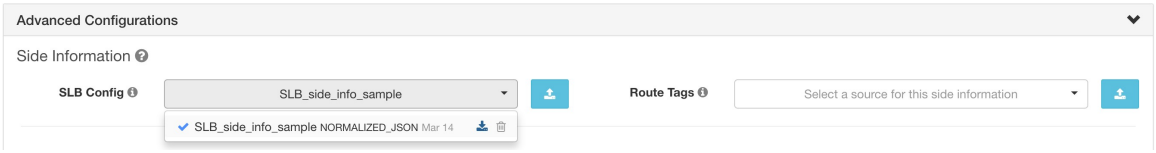
以前にアップロードしたロードバランサ（SLB 構成）またはルートラベルファイルを表示または削除するには、次の手順を実行します。

1. [このサイド情報のソースを選択（Select a source for this side information）] というラベルの付いた個別のボックスをクリックします。

アップロードしたファイルのリストが表示されます。

2. ファイルの横にあるダウンロードアイコンまたはゴミ箱アイコンをクリックして、ファイルを表示または削除します。

図 *273*：アップロードしたサイド情報



クラスタの細分度

クラスタリングの細分度を指定することで、自動ポリシー検出によって生成されるクラスタのサイズを制御できます。

- [細かい（Fine）]：クラスタの数が多くなりますが、サイズが小さくなります。
- [粗い（Coarse）]：クラスタの数は少なくなりますが、サイズは大きくなります。

（注） シスコのアルゴリズムでは、その他の多くのシグナルが考慮に入れられるため、結果に大きな変化が見られない場合があります。たとえば、生成されたクラスタの信頼度が非常に高い場合、このコントロールを変更しても結果はほとんど変わりません。

ポートの一般化

自動ポリシー検出の [詳細設定（Advanced Configurations）] の [ポート一般化（Port Generalization）] オプションは、ポートの一般化（つまり、単一のワークロードでサーバーポートとして使用される多数のポートをポート間隔に置き換えること）を実行するときに必要な統計的有意性のレベルを制御します。

この設定は、ポリシーの精度、数、簡潔さ、およびポリシーの生成に必要な時間に影響を与える可能性があります。

ポートの一般化を無効にするには、スライダを左端に移動します。無効にすると、多数のサーバーポートがワークロードによって使用される場合、自動ポリシー検出および/または自動ポリシー検出の UI レンダリング時間が大幅に遅くなる可能性があることに注意してください。

スライダを右に動かすと、より積極的な一般化が行われます。ポート間隔を作成するために必要な証拠が少なくなり、元のポリシー（単一のポートを含む）をポート間隔に置き換えるための基準も緩和されます。

背景

Hadoop などの一部のアプリケーションは、32000 から 61000 など、一定の間隔で多くのサーバーポートを使用および変更します。自動ポリシー検出は、観察されたフローにおけるワークロードのサーバーポートの使用状況を使用して、各ワークロードでこうした動作を検出しよう

とします。自動ポリシー検出では、可能なポート（ただし100など多数のポート）の合計の一部のみを観察することで、任意のポート（たとえば32000から61000）がワークロードによってサーバーポートとして使用できると「一般化」します。間隔内に含まれるポートは、このような間隔に置き換えられます（最小観測カウントについての特定の基準を満たす場合）。これにより、より少ない、よりコンパクトなポリシーが作成されます。間隔の推定は、正確なポリシーを計算するために重要です。十分な一般化が行われずにポリシーが適用された場合、将来の多数の正当なフローがドロップされてしまいます。多数のポートを1つまたはいくつかの間隔にマージすることにより、UIのレンダリング時間も大幅に高速化されます。

無効化を含む、ポートの一般化の程度を制御することができます。

ポリシー圧縮

ポリシー圧縮が有効になっていて、ワークスペース内の複数のクラスタ内のポリシーが類似している場合、それらのポリシーを親範囲全体に適用可能な1つ以上のポリシーに置き換えることができます。たとえば、ワークスペース内のすべてまたはほぼすべてのクラスタが同じコンシューマに同じポートを提供する場合、それらのクラスタ固有のポリシーはすべて親範囲内の1つのポリシーに置き換えられます。これにより、ポリシーの数を大幅に減らして、可能な限り簡素化することができます。また、ドロップされていた正当なフローを今後は許可できる可能性もあります（正確な一般化）。

圧縮の設定がより積極的であるほど、クラスタ固有のポリシーを親全体に適用できるポリシーに置き換えるために必要なポリシー頻度のしきい値は小さくなります。

範囲ツリーのブランチのポリシーを生成する場合：

このノブを使用して、階層型ポリシーの圧縮の積極性レベルを変更できます。

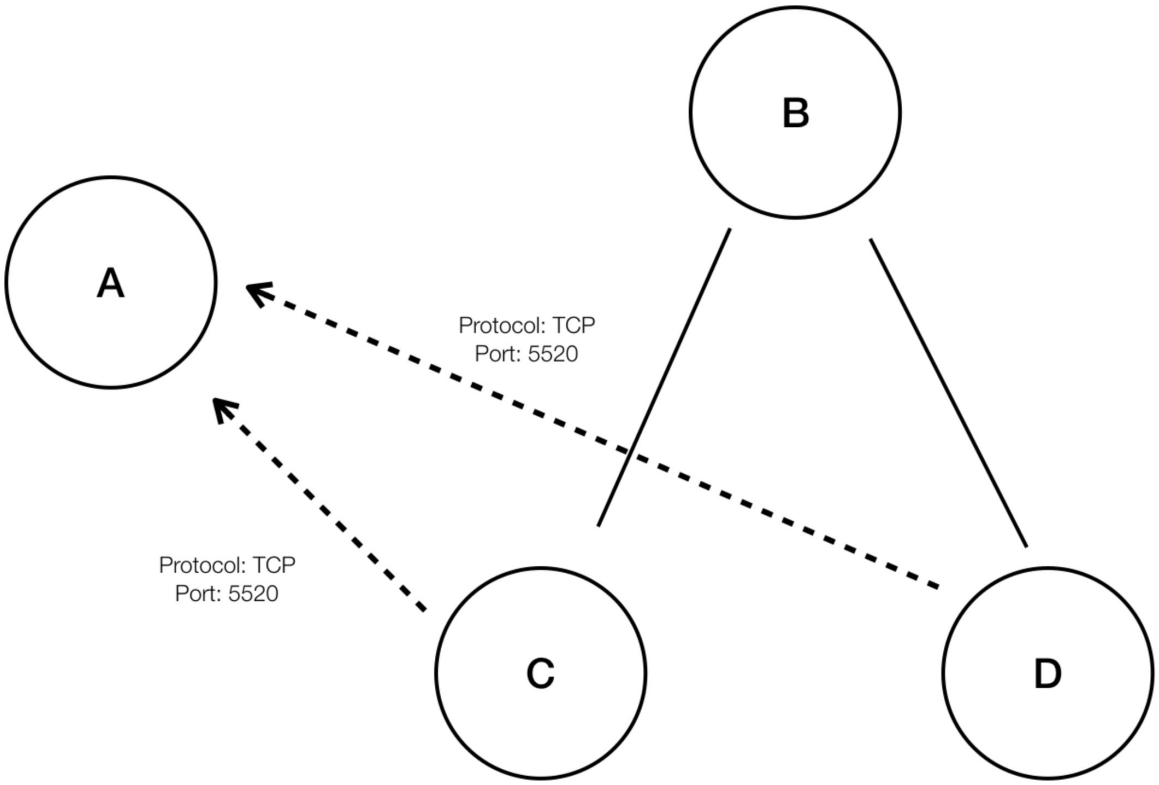
（注） 現在、自動ポリシー検出会話ページは、圧縮ポリシーが導かれた会話の表示をサポートしていません（圧縮を無効にするか、フロー検索を使用する必要がある場合があります）。

階層型ポリシーの圧縮

ポリシーの圧縮は、範囲ツリーのブランチのポリシーを生成するときにも実行できます。ポリシー圧縮ノブを使用して、階層型ポリシー圧縮の積極性レベルを変更できます。階層型ポリシー圧縮の例を以下に示します。

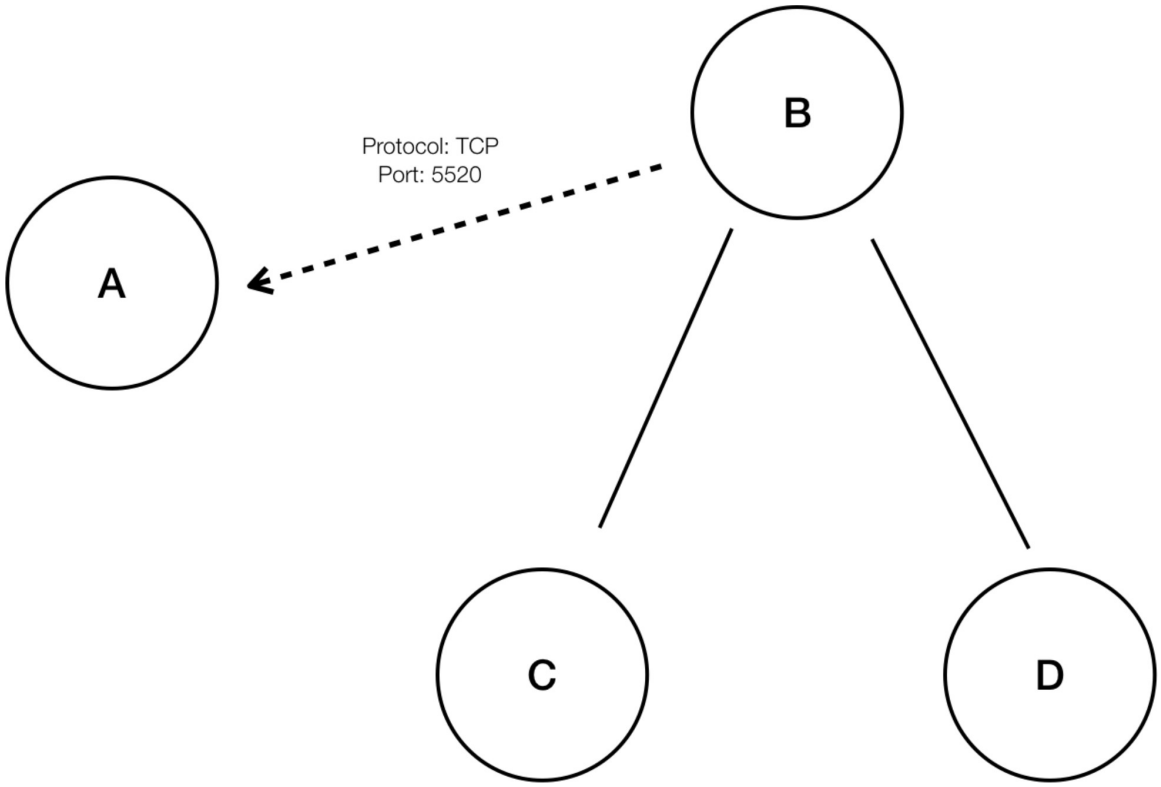
- A、B、C、Dを範囲ツリーの範囲部分とし、「C」と「D」を「B」の子範囲とします。「C」→「A」をポート5520のTCP「ALLOW」ポリシーとし、「D」→「A」をポート5520のTCP「ALLOW」ポリシーとします。

図 *274 :* 階層型ポリシー圧縮前



- 階層型ポリシー圧縮では、十分に大きなグループの子範囲が同じポート、プロトコル、宛先または送信元を共有するポリシーに関係している場合、これらのポリシーを、親範囲から共通の送信元または宛先に接続する一般化されたポリシーに置き換えることができます。上記の場合、「C」と「D」は「B」の子範囲であり、ポリシー「C」→「A」と「D」→「A」は同じ宛先、ポート、プロトコルを共有しています。「B」の子範囲の100%に同様のポリシーが含まれているため、ポリシーは「B」→「A」に昇格され、次のようになります。さらに、階層的な圧縮を繰り返すことができるため、一般化されたポリシーは、サブツリー（範囲ツリーのブランチ）のルートまでたどり着くことができます。

図 *275 :* 階層型ポリシー圧縮後



- ポリシー圧縮ノブを使用すると、ポリシーを共有する子範囲の、圧縮をトリガーする最小必要比率（通常は子範囲の総数に対する割合として測定される）を変更することにより、このような圧縮の積極性を調整できます。無効にすると、各ポリシーは、外部依存関係リストに基づいて、最も優先度の高い範囲間で生成されます。その後、自然に順序付けられた外部依存関係リストを適用することを選択した場合、生成されるポリシーは、数ある範囲の中で最も詳細なポリシーになります。

クラスタリングアルゴリズム（クラスタリングへの入力）

上級ユーザーは、クラスタリングアルゴリズムのデータの主なソース、つまり、ライブネットワークフロー、実行中のプロセス、またはその両方を選択できます。

発信ポリシーコネクタの自動承諾

このオプションは、（上級）クロス範囲ポリシーの作成（616ページ）で説明されている方法を使用して、自動ポリシー検出を使用してクロス範囲ポリシーを作成する場合にのみ適用できます。

自動ポリシー検出中に作成されたすべての発信ポリシー要求は、自動的に受け入れられます。

詳細については、自動承諾ポリシーコネクタ（626ページ）および「ポリシー要求」を参照してください。

（注） このオプションは、ルート範囲の所有者とサイト管理者のみが使用できます。

除外フィルタに一致するフローを無視する

指定したカンバセーションフローを無視するには、該当するオプションを有効にします。いずれかのフィルタリストを表示または変更するには、該当する [除外フィルタ（Exclusion Filters）] リンクをクリックします。詳細については、「除外フィルタ」、「デフォルトの除外フィルタ（565ページ）」および「除外フィルタの構成、編集、または削除（546ページ）」を参照してください。

エージェントのサービス検出の有効化

特定のアプリケーションでは、広範囲のポートを使用するように指定されているものの、実際のトラフィックでは、ポリシー検出に含まれる期間中にこれらのポートのサブセットのみが使用される場合があります。このオプションを使用すると、実際のトラフィックで確認されたポートだけでなく、これらのアプリケーションに指定されたポートのプール全体を、これらのアプリケーションのポリシーに含めることができます。

このオプションを有効にすると、エージェントノードに存在するサービスに関連するエフェメラルポート範囲情報を収集できます。次に、このポート範囲情報に基づいてポリシーが生成されます。

例：

- Windows Active Directory ドメインサーバーは、デフォルトの Windows エフェメラルポート範囲 **49152 ～ 65535** を使用して要求を処理します。このフラグが設定されている場合、このポート範囲情報はエージェントによって報告され、この情報に基づいてポリシーが生成されます。

図 *276*：サービス検出がエージェントで有効になっている

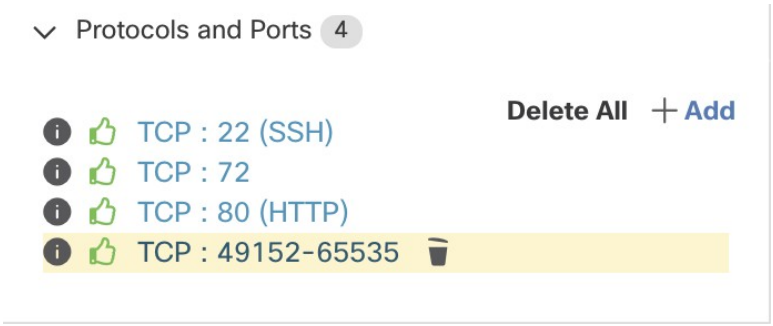
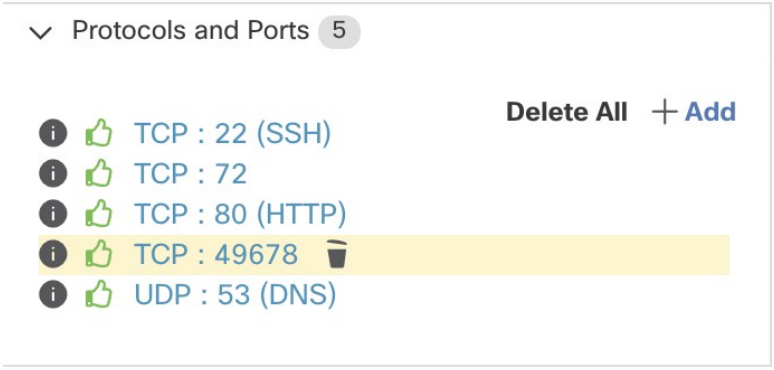


図 *277*: サービス検出がエージェントで有効になっていない



承認されたポリシーの引き継ぎ

このオプションは、デフォルトで有効です。

このフラグが設定されている場合、承認済みとしてマークされたすべてのポリシー（OpenAPIを使用して承認されたものを含む）が保持されます。これにより、自動ポリシー検出が「許可」ポリシーを検出した場合でも効力を発揮する必要がある特定の広範な拒否ルールを再定義する必要がなくなります。

詳細については、承認済みポリシー （566 ページ） を参照してください。

クラスタリングをスキップしてポリシーを作成

このオプションが選択されている場合、新しいクラスタは生成されず、既存の承認済みクラスタまたはインベントリフィルタからポリシーが生成されます。その他にも、ワークスペースに関連付けられた範囲全体が関係します（実際には、範囲全体を単一のクラスタとして扱います）。このオプションを使用すると、ポリシー数が大幅に減る (ただし、粗くなる) 可能性があります。

冗長ポリシー削除の有効化

このオプションは、範囲ツリーのブランチのポリシーを生成する場合にのみ使用できます。

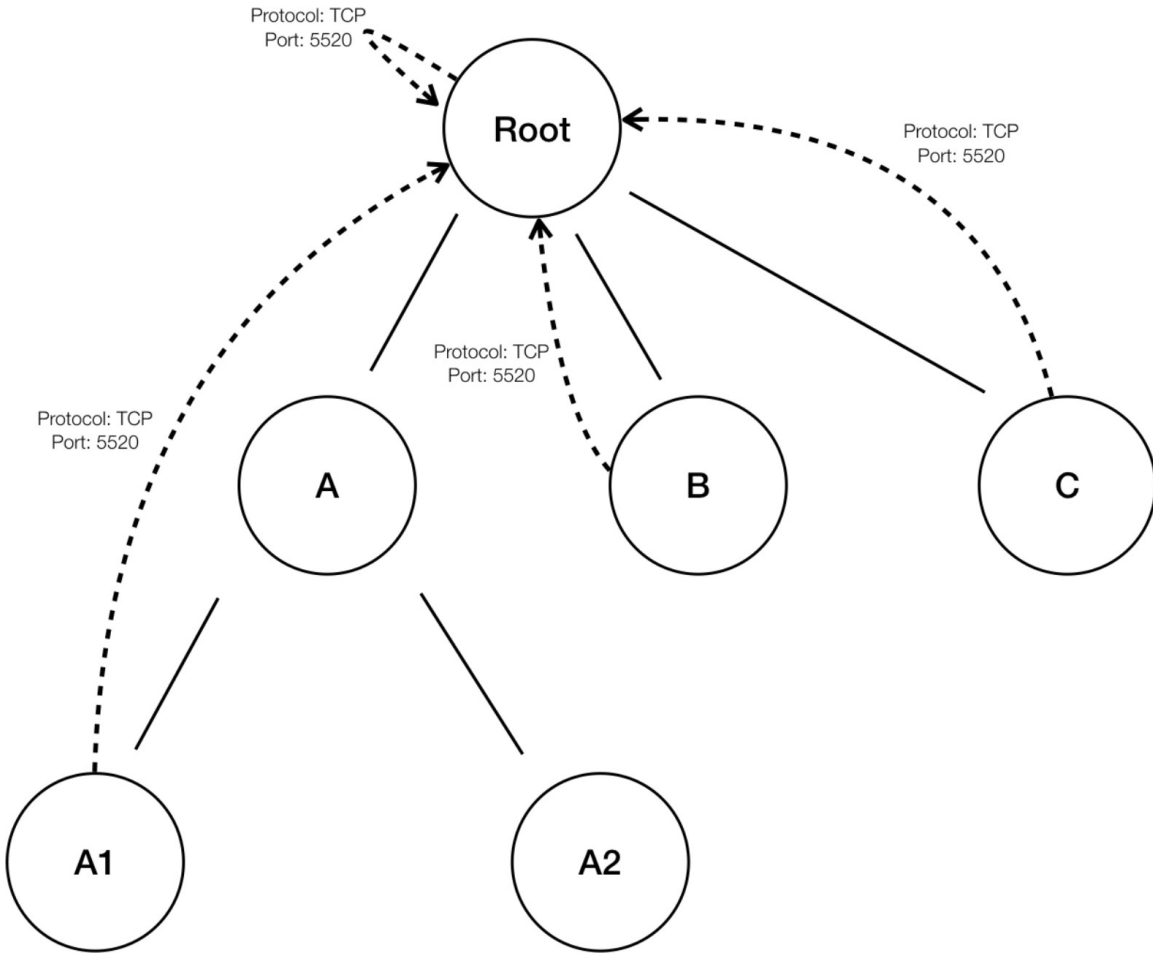
このオプションは、冗長な詳細ポリシーの削除を有効/無効にします。

例：

- ルート、A、B、C、A1、およびA2を範囲ツリーの範囲部分とします。以下をポリシーとします。

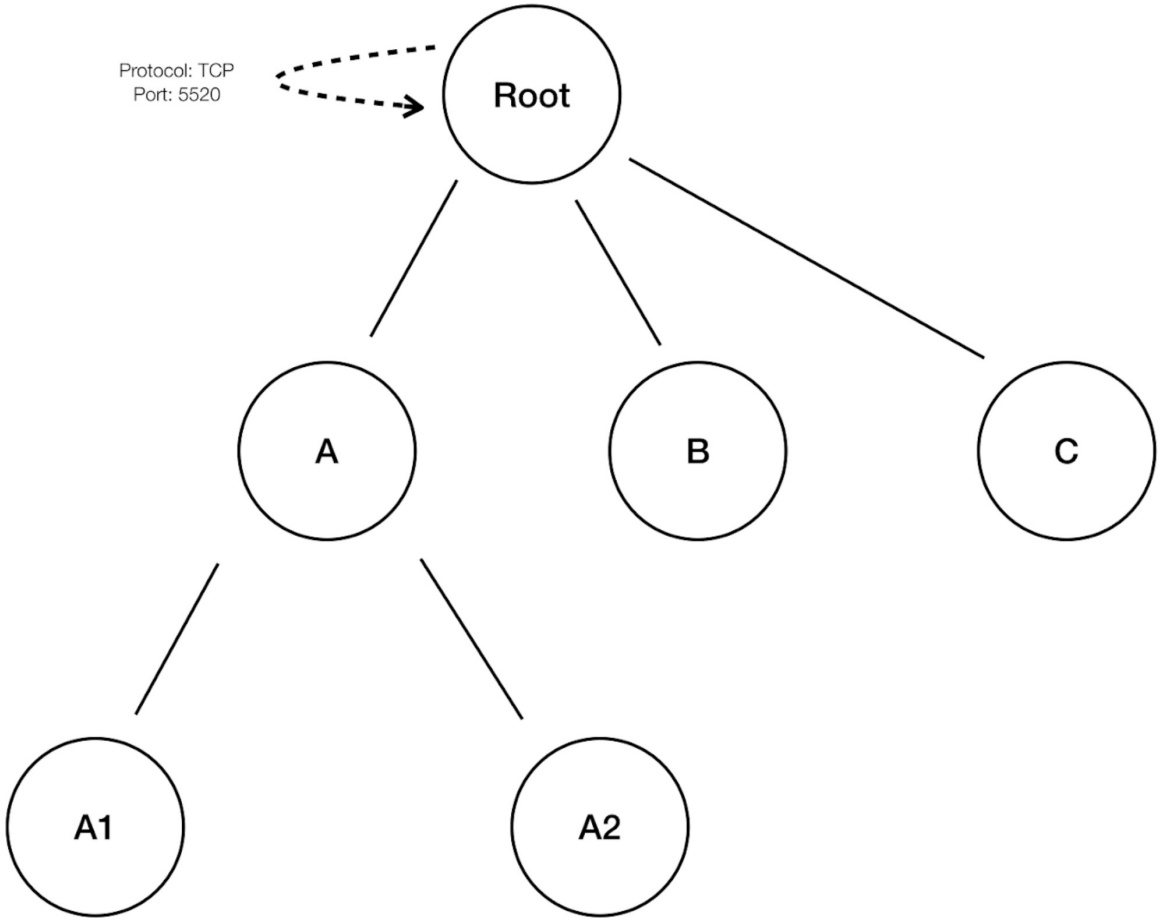
  1. “Root” → “Root”
  2. “B” → “Root”
  3. “C” → “Root”
  4. “A1” → “Root”

図 *278 :* 冗長ポリシーを削除する前



- ポリシー “B” →“Root”、“C” →“Root” および “A1”→ “Root” は、ポリシー “Root”→“Root” がこれらのポリシーをカバーしているため、冗長です。冗長ポリシーの削除機能は、そのようなポリシーをチェックして削除し、次のような単一のポリシー “Root” → “Root”のみを作成します。

図 *279 :* 冗長ポリシーを削除した後



冗長ポリシーの削除は、解釈可能なポリシーの簡潔なセットを維持するのに非常に役立ちます。縮小されたポリシーセットには、すべてのワークロードトラフィックをカバーするために選択された圧縮レベルで、最小限の数のポリシーが含まれます。ただし、ポリシー分析を通じてポリシーを常に監査し、対応するカンバセーションを調べて、結果として得られるポリシーの厳しさを評価する必要があります。より細かい範囲またはインベントリフィルタに分類されていないエンドポイントとの間のトラフィックが存在する場合、これは特に重要です。このようなエンドポイントは、ルート範囲を含むポリシーなど、意図したよりも粗いポリシーの生成を引き起こす可能性があります。それと同時に冗長ポリシーの削除が有効になっている場合、より粒度の高いポリシーは削除され、表示されなくなります。（圧縮された）ポリシーのソースを診断し、より細かいレベルのポリシーを確認するには、ポリシーの圧縮と冗長ポリシーの削除機能をオフにします。また、現在、自動ポリシー検出のカンバセーションページでは、圧縮/一般化されたポリシーにつながるカンバセーションが表示されない場合があることにも注意してください。これを回避するには、圧縮と冗長ポリシーの削除を無効にすると、生成されたポリシーにつながるカンバセーションを見つけやすくなります。

ヒント　範囲ツリーのブランチのポリシーの検出は、ワークスペース範囲をルートとする範囲サブツリーのすべてのポリシーを検出するため、これらのポリシーは、サブツリーの下のすべてのワークロードの自動ポリシー検出によって検出されるすべての合法的なトラフィックをカバーします。ポリシー分析（「ポリシー」を参照）などのツールを使用してこれらのポリシーを分析する場合は、サブ範囲に関連付けられたすべてのワークスペースでポリシー分析をオフにする必要があります。こうすると、（通常、より具体的な範囲定義のために高い優先度に設定される）サブ範囲ワークスペースに存在するポリシーがあっても優先されず、結果に干渉しません。ただし、サブ範囲ワークスペースのポリシーが、通常、サブ範囲に固有のより細かいインベントリフィルタまたはクラスタを含むさまざまなトラフィックセットをカバーするように設定されている場合は、例外が適用されます。

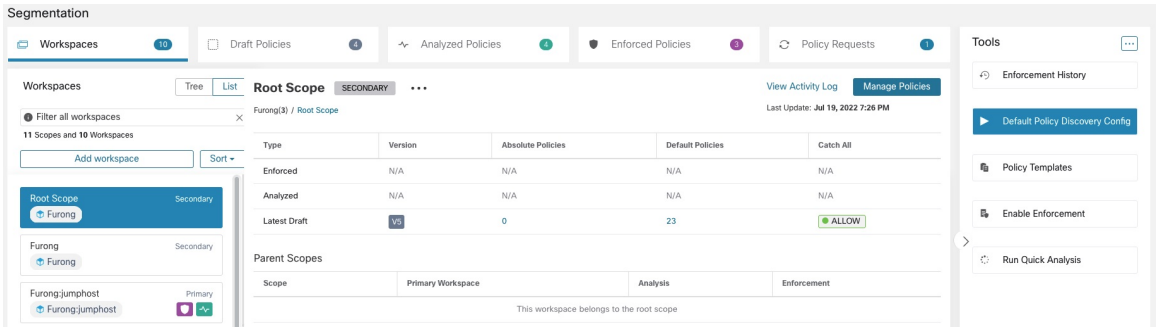
デフォルトのポリシー検出設定

ルート範囲全体の任意のワークスペースでオプションで使用できる、デフォルトの自動ポリシー検出設定を設定することができます。

ポリシー検出のデフォルトオプションを設定するには、次の手順を実行します。

[防御（Defend）]>[セグメンテーション（Segmentation）]を選択し、ページの右側にあるキャレット記号をクリックして [ツール（Tools）] メニューを展開します。次に、[デフォルトのポリシー検出設定（Default Policy Discovery Config）] を選択します。

図 *280 : [デフォルトのポリシー検出設定（Default Policy Discovery Config）] ページへの移動*



[デフォルトのポリシー検出設定（Default Policy Discovery Config）] ページのオプションについては、以下を参照してください。

- 外部依存関係 （549 ページ） およびサブトピック
- 自動ポリシー検出の詳細設定 （554 ページ） およびサブトピック
- デフォルトの除外フィルタ （565 ページ）

重要　デフォルト設定が完了し、個々のワークスペースで使用する準備ができたら、[保存（Save）] をクリックします。

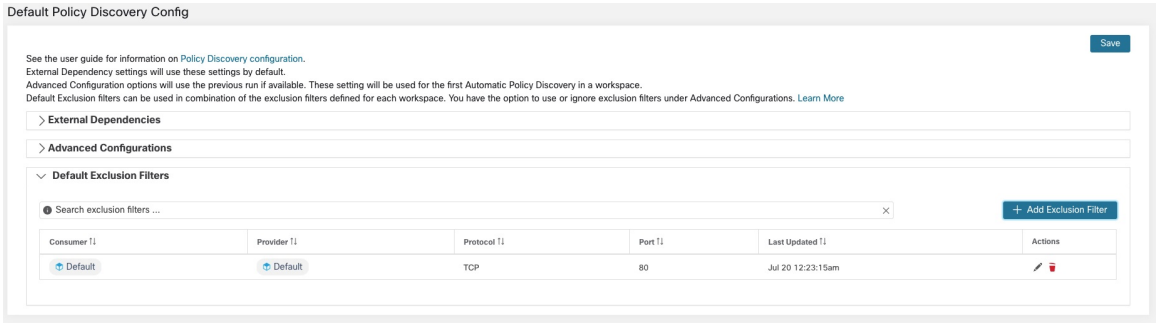
デフォルトの除外フィルタ

除外フィルタを使用して、検出対象外とするトラフィックフローを指定することで、自動ポリシー検出によって提案されたポリシーとクラスタを微調整できます。

詳細については、「除外フィルタ」を参照してください。

テナント内のすべてのワークスペースで使用できるグローバルなデフォルトの除外フィルタリストを作成し、ポリシーを検出するときにこのデフォルトリストを使用するかどうかをワークスペースごとに指定できます。

図 *281* : デフォルトの除外フィルタ



デフォルト除外フィルタの設定方法については、「除外フィルタの構成、編集、または削除（546 ページ）」を参照してください。

デフォルトの除外フィルタを有効または無効にする方法については、除外フィルタを有効または無効にする （549 ページ）を参照してください。

高度なポリシー検出設定を実現するためのロードバランサ設定の取得

サポートされているロードバランサ設定ファイルを Secure Workload に直接アップロード可能な形式で取得し、ポリシー検出で使用できるようにするための手順を以下で紹介します。詳細については、「自動ポリシー検出の詳細設定」および「ポリシーの検出時にロードバランサとルータからのデータを含める （554 ページ）」を参照してください。

すべてのファイルは ASCII としてエンコードする必要があることに注意してください。

Citrix Netscaler

コンソールで `show run` の出力を連結し、ファイルをアップロードします。

「サンプル設定ファイル」を参照してください。

F5 BIG-IP

`bigip.conf` ファイルをアップロードします。.UCS 拡張子の付いたファイルがある場合は、アーカイブを解凍し、設定ダンプ内の `bigip.conf` ファイルのみをアップロードします。複数のファイルがある場合は、それらを連結してアップロードします。

「サンプル設定ファイル」を参照してください。

HAProxy

`haproxy.cfg` ファイルをアップロードします。通常、パスは `/etc/haproxy/haproxy.cfg` です。

「サンプル設定ファイル」を参照してください。

正規化された *JSON*

上記のオプションでは限定的だと思われる場合は、設定を次の JSON スキーマに変換し、それらを直接アップロードしてください。JSON ファイルのサンプルは、自動ポリシー検出用の [詳細な実行設定（Advanced Run Configurations）] の [SLB設定（SLB Config）] の横にある **i** アイコンをクリックして直接ダウンロードできます。

「サンプル設定ファイル」を参照してください。

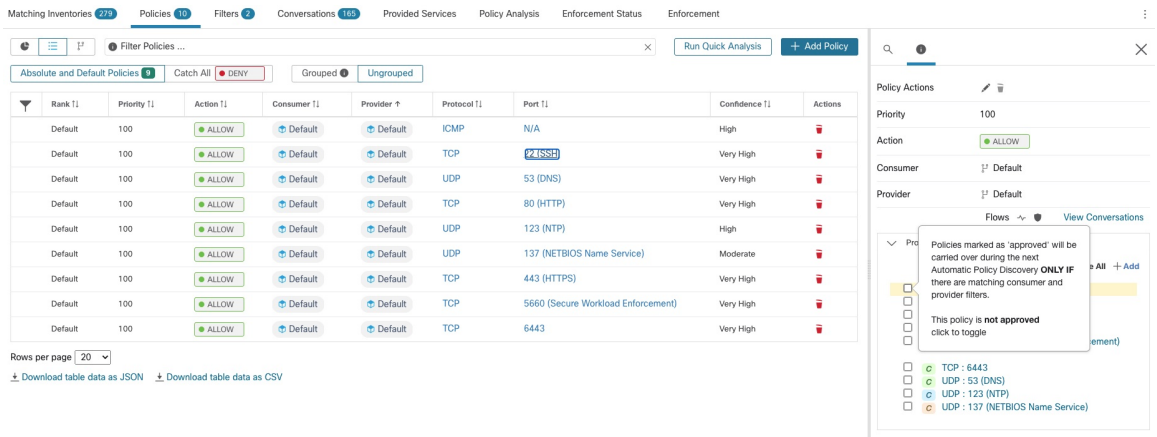
ポリシーの承認

ポリシー検出の結果を確認しているときに、検出されたポリシーのうち保持したいものを承認すると、その後ポリシーを検出したときにそのポリシーがそのまま引き継がれます。詳細については、承認済みポリシー （566 ページ）を参照してください。

ポリシーを承認するには、次の手順を実行します。

1. [ポリシー（Policies）] ページで、保護するポリシーの [プロトコルとポート（Protocols and Ports）] 列の値をクリックします。
2. 右側に表示されるパネルで、今後のポリシー検出時にポリシーを保持する各プロトコルとポートの左側にあるチェックボックスを選択します。

図 *282 :* ポリシーの承認



この手順を使用して、ポリシーから承認を削除することもできます。

承認済みポリシー

一般的に、承認済みポリシーは自動ポリシー検出中に変更されず、自動ポリシー検出では、承認済みポリシーの効果と重複するポリシーは提案されません。

承認されたポリシーは次のとおりです。

- 手動で作成されたポリシー。
- 手動で承認された検出済みのポリシー。

  （ポリシーが意図したとおりに動作することを確認したら、ポリシーを承認して将来の自動ポリシー検出中に変更されないようにします。ポリシーの承認（566ページ）を参照してください。）
- アップロードされたポリシー（明示的に `approved: false` とマークされていない場合に限ります）。
- 親および先祖の範囲で（特に、プライマリワークスペースの最新バージョンから）定義された、この範囲内のワークロードに適用される承認済みポリシー。
- コンシューマとプロバイダーが異なる範囲にある場合：ポリシーオプション（615ページ）で説明されている高度な方法を使用してクロス範囲ポリシーを処理する場合に、別のワークスペースからのポリシーリクエストが受け入れられたときに作成されるポリシー。たとえば、提供されるサービス（628ページ）タブから含まれるポリシーが含まれます。

ポリシーのポートやプロトコルのリンクをクリックして、ページの右側のパネルに詳細を表示すると、プロトコルタイプの横にある親指アイコンで承認済みポリシーが示されます。

承認済みポリシー保護の例外

ポリシーの両端が、承認済みクラスタ、インベントリフィルタ、受け入れられたポリシー要求（クロス範囲ポリシーの場合）、またはメンバーシップを大幅に変更しないクラスタのいずれかである場合、承認済みポリシーは将来の自動ポリシー検出時にも保持されます（ただし、最後のケースではクラスタメンバーシップが変更されている可能性があります）。

ポリシーのいずれかの端が非承認クラスタであり、自動ポリシー検出時に、そのようなクラスタと十分に重複している新しく生成されたクラスタがない場合、承認済みポリシーは、将来の自動ポリシー検出の実行中に保護されない可能性があります。

未承認のクラスタを含むポリシーを保護するには、ポリシーの両端でクラスタを明示的に承認する必要があります。

また、デフォルトで有効になっている自動ポリシー検出の詳細設定もあります。承認済みポリシーを変更から保護しない場合は、ワークスペースまたはグローバルのデフォルトポリシー検出設定についてこのオプションの選択を解除できます。承認されたポリシーの引き継ぎ（561ページ）を参照してください。

承認されたポリシーのトラブルシューティング

承認済みポリシーが引き継がれません

承認済みポリシーが期待どおりに引き継がれない場合は、自動ポリシー検出の詳細設定またはデフォルトの構成設定で、[承認済みポリシーの引き継ぎ（Carry over approved policies）] オプションが選択されていることを確認してください。

ポリシー生成から除外されるカンバセーションを探す

自動ポリシー検出中に、既存の承認済みポリシーの基準に一致するカンバセーションは、ポリシー生成から除外されます。この省略により、同じカンバセーションをカバーする冗長なポリシーが生成されなくなります。（このプロセスは、ポリシーの代わりに一致フィルタを定義する除外フィルタとは異なります（「除外フィルタ」を参照）。除外フィルタは、一致するカンバセーションが自動ポリシー検出のあらゆる部分で表示されないようにします）

これらのカンバセーションから冗長ポリシーは生成されませんが、自動ポリシー検出がクラスタを分析して生成するときに、カンバセーションは引き続き考慮されることに注意してください。

既存の承認済みポリシーによって自動ポリシー検出から除外されているカンバセーションを確認するには、次の手順を実行します。

[カンバセーション（Conversation）] ビュー（「カンバセーション」を参照）で、除外フラグを使用してカンバセーションをフィルタリングします。また、ページの右側に表示されるポリシーの詳細ビューで、これらのカンバセーションを除外する既存の承認済みポリシーの結果を調べることもできます。これを行うには、ポリシー内のポートとプロトコルのリンクをクリックして、カンバセーションの横にある除外アイコンをクリックします（適切なアイコンを見つけるには、アイコンにカーソルを合わせます）。

反復的なポリシーの変更

単一の範囲およびネットワーク全体のポリシーの定義と調整は、反復的なプロセスになります。

検出されたポリシーと手動で作成されたポリシーの両方を変更できます。

自動ポリシー検出の再実行

自動ポリシー検出はいつでも再実行できます。自動ポリシー検出を再実行する最大の理由は、前回の実行時に含まれていなかった追加情報を含めること、または役に立たない情報を除外することです。たとえば、以下を行うことができます。

- 追加のエージェントをインストールするか、追加のコネクタを設定して、一部のフローデータが蓄積されるようにする。
- 検出に使用する期間を長くして、より多くのデータを含める。
- （最初に編集するかどうかに関係なく）クラスタを承認する。これにより、再実行時に他のワークロードのクラスタリングを改善できます。クラスタの承認 （605 ページ）を参照してください。
- ポリシーに影響を与えないことがわかっているフローを除外して、編集の必要をなくす。除外フィルタ （545 ページ）を参照してください。
- 詳細設定を変更する（詳細については、自動ポリシー検出の詳細設定 （554 ページ）を参照してください）。
- ポリシーの複雑さの対処 （607 ページ）に変更を加えた後に変更をキャプチャする。

既存のワークスペースでポリシーを再度自動検出すると、ワークスペース内に異なるクラスタとポリシーが生成される場合があります。

ホストがワークスペースの範囲内になくなった場合、その後の自動ポリシー検出の実行時に、そのホストはどのクラスタにも表示されません。ホストが承認されたクラスタ内にあった場合、そのクラスタに表示されなくなります。時間枠または設定が異なる同じメンバーワークロードのセットであっても、自動ポリシー検出によって異なるクラスタが生成される場合があります。



（注） ポリシー検出中に変更されないポリシーのタイプのリストについては、承認済みポリシー（566ページ）を参照してください。



（注） 冗長ポリシーの削除後続の自動ポリシー検出では、プライマリワークスペースで承認されたポリシーによって、ポリシー生成のために一致するカンバセーションが削除されるため、冗長ポリシーは生成されません。除外フィルタの場合と同様に、ポリシーが非プライマリワークスペースで定義されたクラスタフィルタを使用している場合、この機能は非プライマリワークスペースでは完全に機能しない可能性があることに注意してください。非プライマリワークスペースからのクラスタフィルタはアクティブではなく、どのフローにも一致しないため、自動ポリシー検出中に非プライマリワークスペースで冗長ポリシーが引き続き生成される可能性があります。

重要：自動ポリシー検出を再実行する前に

重要 ワークスペースでポリシーを再検出する前に、次の各項目に対処します。

- デフォルトでは、特定のワークスペースでポリシーを検出するたびに、以前に検出されたポリシーとクラスタのセットは、新しい検出期間に含まれているデータに基づいて上書きされます。一部のポリシーとクラスタを保持し、他は保持しない場合は、保持するポリシーとクラスタを承認します。
- 生成された既存のクラスタを保持する場合は、「自動ポリシー検出の再実行中のクラスタ変更の防止」またはクラスタの承認 （605 ページ）を参照してください。
- 既存の生成されたポリシーを保持する場合は、ポリシーの承認 （566 ページ） を参照してください。
- 変更しない限り、以前の検出実行で設定された既存の [詳細設定（Advanced Configuration）] の設定が使用されます。

  ただし、設定されたデフォルトの外部依存関係は、以前の実行の依存関係よりも優先して使用されます。
- 検出されたポリシーの現在表示されているバージョンが最新バージョンではなく、以前に検出されたバージョンを保持する場合は、ページの上部に表示されているバージョンをクリックし、最新の v* バージョンを選択します。

  以前のバージョンが表示されている場合は、そのバージョンと新しく検出されたバージョンの間のバージョンが削除されます。

  詳細は、検出されたポリシーバージョンの表示、比較、および管理 （570 ページ）を参照してください。

ポリシー検出を再実行するには、ポリシーの自動検出 （543 ページ）を参照してください。このトピックの項目に対処した後は、ポリシーを検出するたびに同じプロセスになります。

検出されたポリシーバージョンの表示、比較、および管理

ワークスペースでポリシーを検出するたびに、一連のポリシーに割り当てられたバージョン番号（v*）が増加します。

詳細については、ポリシーバージョン（v* および p*）について （673 ページ） を参照してください。

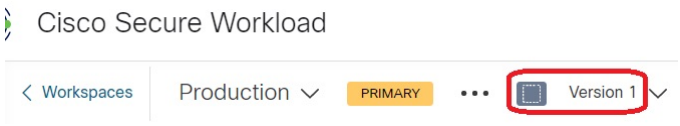
手順

ステップ 1 [防御（Defend）] > [セグメンテーション（Segmentation）] をクリックします。

ステップ 2 ワークスペースに移動します。

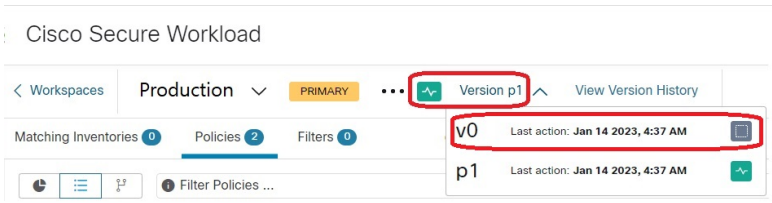
ステップ 3 [ポリシーの管理（Manage Policies）] をクリックします。

ステップ 4 自動ポリシー検出によって生成された、現在表示されているポリシーのバージョンがページの上部に表示されます。



すでにポリシーを分析または適用している場合、表示されるバージョンには、ポリシー検出バージョン、分析されたポリシーバージョン、または適用されたバージョンがあります。

ステップ 5 次のいずれかを実行します。

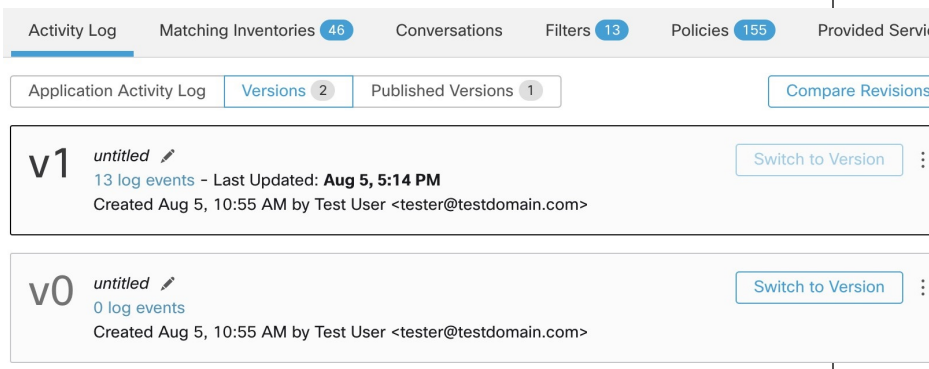
自動ポリシー検出によって生成された別のバージョンのポリシーを表示する。	現在のバージョンをクリックし、別の v* バージョンを選択します。 （p* バージョンが表示されている場合、それらは検出されたポリシーのバージョンではなく、分析されたバージョンや適用されたバージョンです）。  **重要**：この手順の最後にある「次の作業」の項の注意事項を参照してください。

バージョンに関する詳細を表示する。	

1. ページ上部の現在のバージョンの横にある [バージョン履歴の表示（View Version History）] をクリックします。
2. [バージョン（Versions）] タブをクリックして、検出されたポリシーのバージョンを表示します（[公開されたバージョン（Published Versions）] タブではありません）。

   バージョンのリストが表示されます。

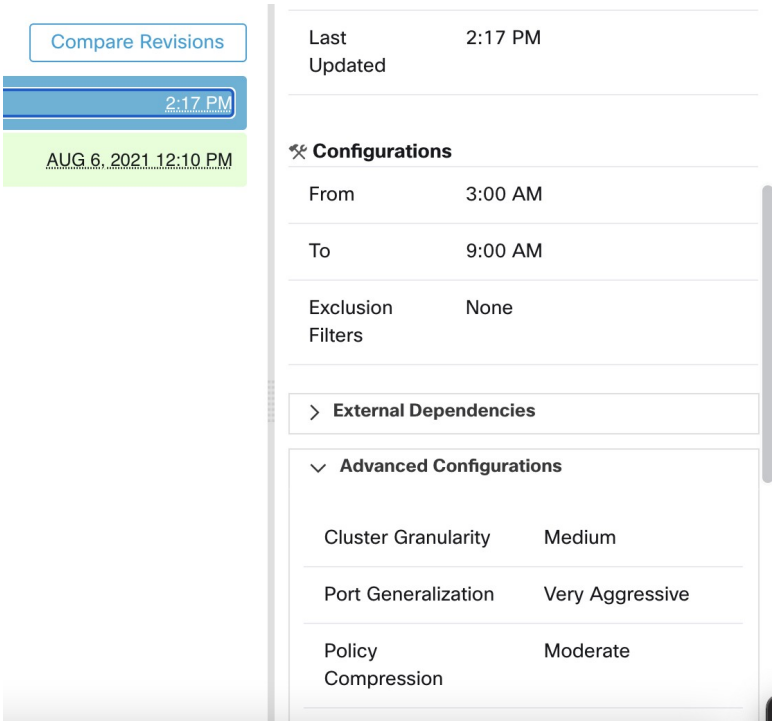
   図 *283 :* 概要情報を含む生成されたポリシーバージョンのリスト



3. バージョンの [ログイベント（log events）] リンクをクリックします。
4. イベント行のリンクをクリックします。

   使用可能な詳細には、統計、除外フィルタ、外部依存関係、および実行の設定があります。

   図 *284 :* 特定の自動ポリシー検出の実行に使用される設定

	
2 つのバージョンを比較して変更点を確認する。	**1.** [リビジョンの比較（Compare Revisions）] をクリックします。 **2.** 比較するバージョンを選択します。 **3.** 結果の詳細については、ポリシーバージョンの比較：ポリシーの差分（676 ページ）を参照してください。
不要なバージョンを削除する。	バージョンの ⋮ をクリックし、[削除（Delete）] を選択します。 自動ポリシー検出によって生成された（v* バージョンの）最後に残っているバージョンは、削除できません。
バージョンをエクスポートする。	バージョンの ⋮ をクリックし、[エクスポート...（Export...）] を選択します。

次のタスク

重要 検出されたポリシーの以前のバージョンを保持する場合は、古いバージョンでの作業が完了したときに、検出されたポリシーの現在のバージョンを常に表示します。

次にこのワークスペースのポリシーを検出したときに、検出されたポリシーの最新バージョンが表示されていない場合は、古いバージョンが削除される可能性があります。

たとえば、検出されたポリシーの最新バージョンが v4 であり、ポリシーを再度検出したときに v2 が表示されている場合、既存の v3 と v4 が削除され、新しく検出されたポリシーのバージョンは v3 になります。

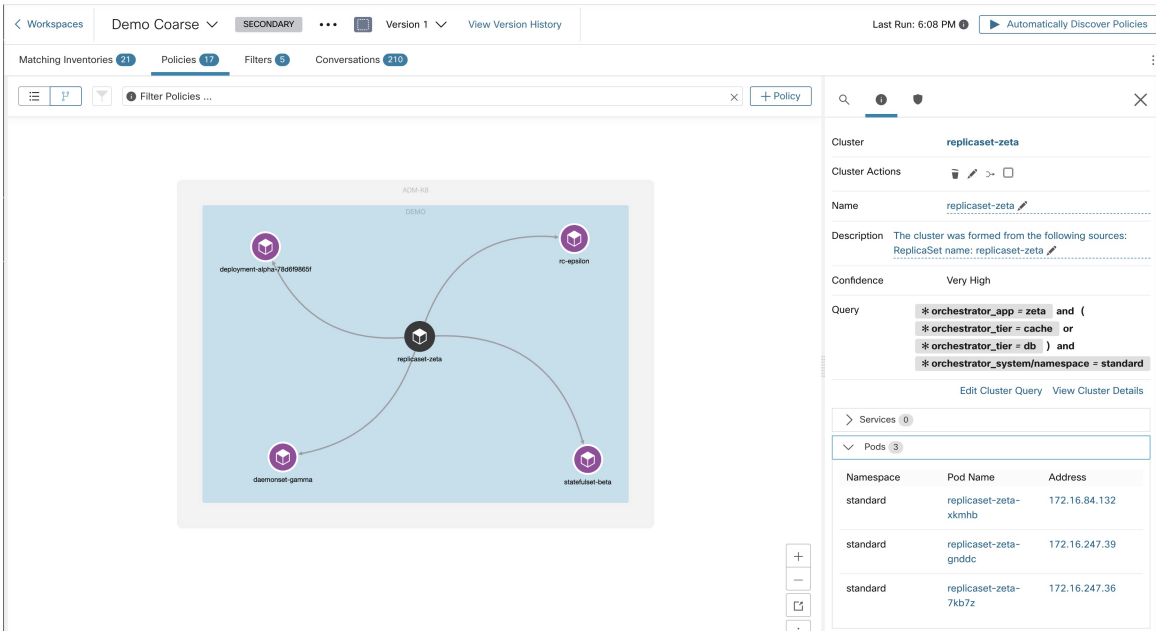
この動作により、バージョン履歴が直線的になり、必要に応じて以前のバージョンに簡単に戻すことができます。

さらに、最新の v* バージョンが表示されている場合にのみ、ポリシーを手動で作成することができます。

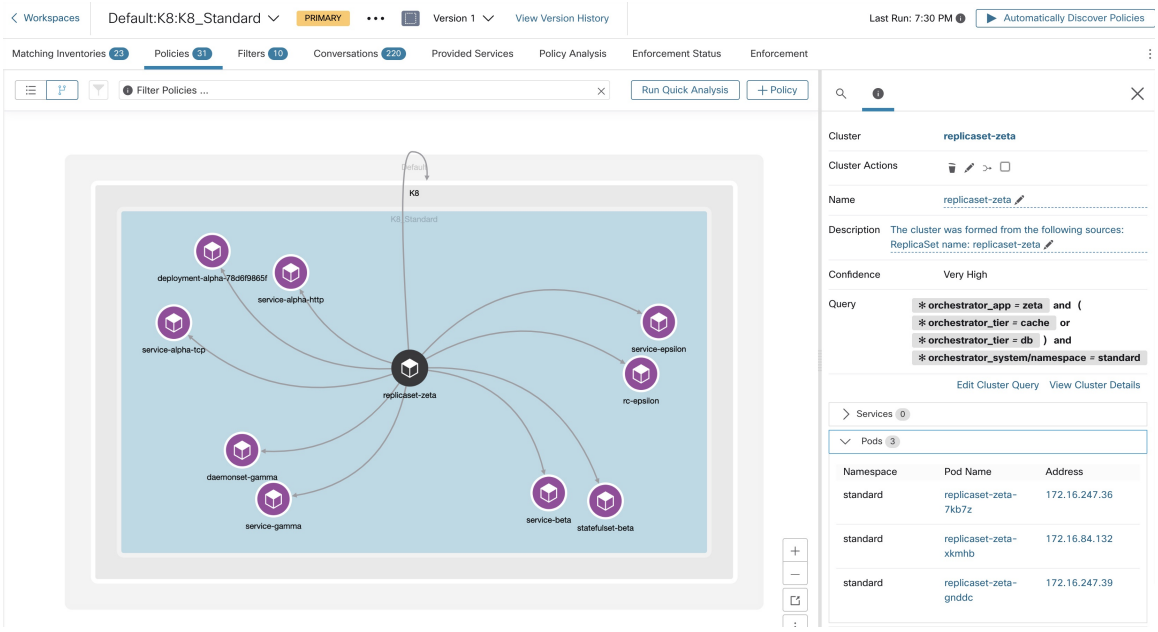
ポリシー検出の Kubernetes のサポート

ポリシー検出では、Kubernetes 設定のポッドとサービスに関する情報が使用されて、ポッドとサービス両方のクラスタが作成され、それぞれのポリシーが生成されます。

クラスタ粒度が粗いまたは非常に粗いに設定されている場合、サービスとそれらを支援するポッドが一緒にクラスタ化されます。



クラスタ粒度が、中、細かい、または非常に細かいに設定されている場合、サービスとそれらを支援するポッドが個別にクラスタ化されます。



ポッドクラスタの場合は、ソース情報がクラスタの説明の一部として追加され、説明の各クラスタには、クラスタが形成される原因となったエンティティの情報が含まれます。

たとえば、[説明（Description）]：「このクラスタは次のソースから形成されました：ReplicaSet 名前：replicaset-zeta（The cluster was formed from the following sources: ReplicaSet name: replicaset-zeta）」などです。

インポート/エクスポート

ワークスペースのエクスポート

各ワークスペースのクラスタとポリシーに関連するすべてのコンテンツは、JSON、XML、YAML などの一般的な構造化ドキュメント形式の 1 つのファイルとしてダウンロードできます。これらのファイルを使用して、社内でさらに処理したり、他のポリシー適用や分析ツールで取り込んだりできます。

ワークスペースヘッダーの **[. . .]** メニュー項目に移動し、[エクスポート（Export）] 項目をクリックすると、エクスポートダイアログが表示されます。エクスポートされたファイルに、実際のネットワークフローに基づいた自動ポリシー検出によって生成されたクラスタ間のセキュリティポリシーおよびクラスタコンテンツを含めるか、クラスタコンテンツのみを含めるかを選択できます。目的の形式を選択し、[ダウンロード（Download）] をクリックして、ファイルをローカルファイルシステムにダウンロードします。

図 *285 : [*インポート/エクスポート（*Import/Export*）*]* メニュー項目

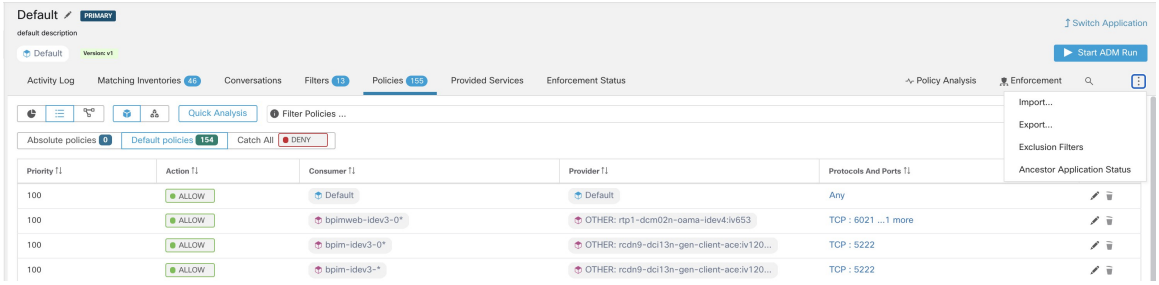
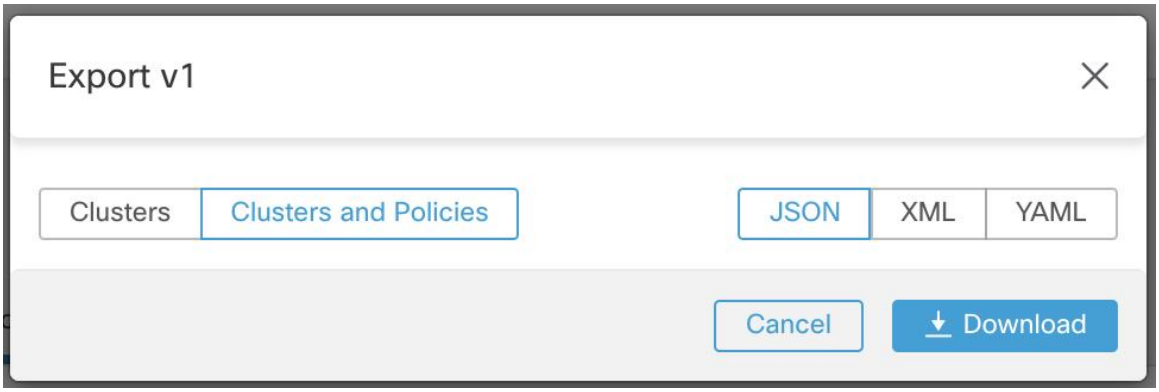


図 *286 :* ワークスペースのポリシーのエクスポート



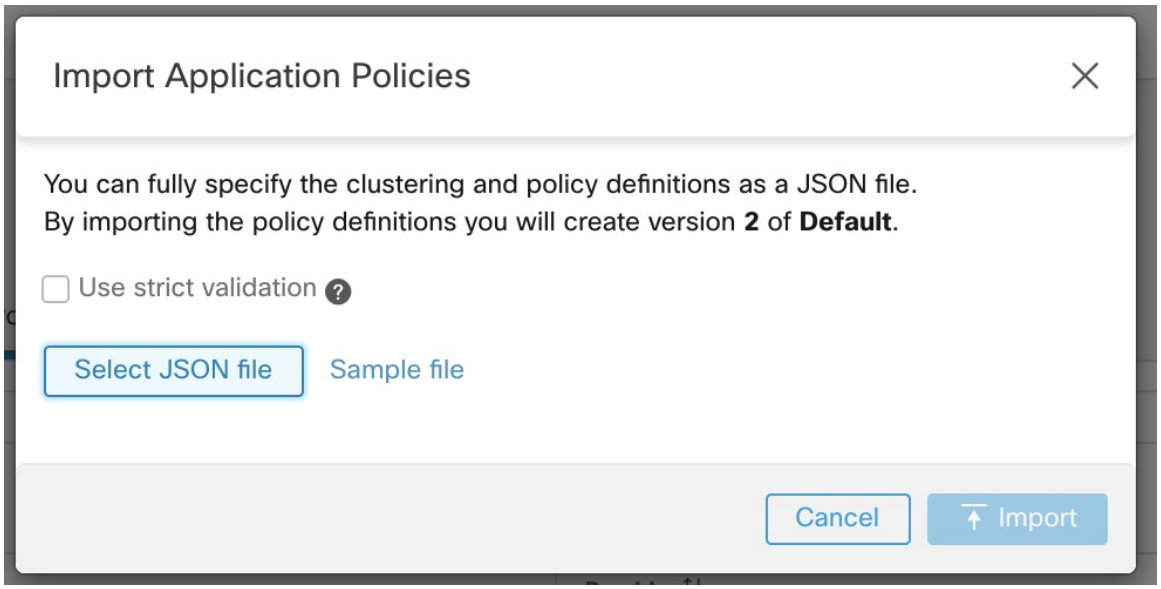
ワークスペースをエクスポートすると、自動ポリシー検出構成の [発信ポリシーコネクタの自動承諾（Auto accept outgoing policy connectors）] 設定が含まれ、インポートされたワークスペースでアクティブになります。

Import

JSON ファイルを直接アップロードすることで、既知のクラスタとポリシーの定義をワークスペースにインポートできます。自動ポリシー検出と同様に、ポリシーを既存のワークスペースにアップロードすると、新しいバージョンが作成され、クラスタとポリシー定義が新しいバージョンの下に配置されます。欠落しているフィルタと正しくないプロパティ値がある場合、エラーが返されます。

ワークスペースヘッダーの **...** メニューのメニュー項目 [インポート（Import）] をクリックします。インポートダイアログで、有効な形式の JSON ファイルを選択できます。[サンプル（Sample）] ボタンをクリックすると、ポリシーとクラスタのスキーマを示す小さいサイズのサンプル JSON ファイルが見つかります。

図 *287* : クラスタ/ポリシーのインポート



厳密な検証 有効にすると、JSON に認識されない属性が含まれている場合にエラーが返されます。このオプションは、タイプミスや間違って識別されたオプションフィールドを見つけるのに役立ちます。

（注） 明示的に承認済みとして表示されていない限り、インポートされたすべてのポリシーはデフォルトで `approved: false` として表示されます。新しいポリシーセットを生成するための自動ポリシー検出中に、この種の承認済みポリシーを保持し続けるオプションがあります。詳細な情報については、「承認済みポリシー （566 ページ）」を参照してください。

専門的なヒント：アプリケーション ワークスペースまたはをエクスポートすることによって取得される JSON ファイルのスキーマには、ポリシーをワークスペースにインポートするために必要な形式とのスキーマ互換性があります。したがって、エクスポートとその後のインポートを使用して、あるアプリケーション ワークスペースから別のアプリケーション ワークスペースにポリシーを複製できます。ポリシーをエクスポートしてからインポートすると、多くの機能が同じように動作しない場合があることに注意してください。たとえば、ポリシーを支援するカンバセーションはエクスポートに含まれず、ポリシーのインポート時にも存在しません。

プラットフォーム固有のポリシー

エージェントが各プラットフォームでポリシーを適用する方法に関する重要な詳細については、エージェントによるポリシーの適用 （61 ページ）を参照してください。Kubernetes/OpenShift については、コンテナへの適用 （663 ページ）を参照してください。

Windows

Windows OS ベースの推奨ポリシー設定

可能な場合は、常にポリシーでポートとプロトコルを指定します。任意のポート、任意のプロトコルを許可することはお勧めしません。

たとえば、ポートとプロトコルの制限を含めて生成されたポリシーは次のようになります。

```
 dst_ports {
  start_port: 22
  end_port: 22
  consumer_filters {
    application_name: "c:\\test\\putty.exe"
  }
 }}
 ip_protocol: TCP
```

対照的に、任意のプロトコルと任意のポートを使用して iperf.exe によって開始されたネットワーク接続を許可する場合、生成されるポリシーは次のようになります。

```
match_set {
  dst_ports {
    end_port: 65535
    consumer_filters {
      application_name: "c:\\test\\iperf.exe"
    }
  }
  address_family: IPv4
  inspection_point: EGRESS
  match_comment: "PolicyId=61008290755f027a92291b9d:61005f90497d4f47cedacb86:"
}
```

上記のフィルタの場合、Secure Workload はプロバイダーのネットワークトラフィックを許可するポリシールールを次のように作成します。

```
match_set {
    dst_ports {
    end_port: 65535
  }
  address_family: IPv4
  inspection_point: INGRESS
  match_comment: "PolicyId=61008290755f027a92291b9d:61005f90497d4f47cedacb86:"
}
```

このネットワークルールは、プロバイダーのすべてのポートを開きます。任意のプロトコルを使用して OS ベースのフィルタを作成しないことをお勧めします。

Windows 属性のポリシーの設定

Windows ベースのワークロードにポリシーを適用する際の精度をさらに高めるために、次の方法でネットワークトラフィックをフィルタリングできます。

- アプリケーション
- サービス名（Service Name）

- ユーザー名（ユーザーグループありまたはなし）

これは、WAF および WFP モードでサポートされています。Windows OS ベースのフィルタは、生成されたネットワークポリシーでコンシューマフィルタとプロバイダーフィルタに分類されます。コンシューマフィルタは、コンシューマワークロードで開始されたネットワークトラフィックをフィルタリングします。プロバイダーフィルタは、プロバイダーワークロード宛てのネットワークトラフィックをフィルタリングします。

始める前に

この手順は、既存のポリシーを変更していることが前提となります。Windows OS ベースのフィルタを追加するポリシーを作成していない場合は、最初にそのポリシーを作成します。



重要 Windows 属性を含むポリシーについては、警告 （74 ページ）および既知の制限事項 （74 ページ）を参照してください。

手順

ステップ 1 [防御（Defend）] > [セグメンテーション（Segmentation）] の順に選択します。

ステップ 2 Windows OS ベースのフィルタを設定するポリシーを含む範囲をクリックします。

ステップ 3 ポリシーを編集するワークスペースをクリックします。

ステップ 4 [ポリシーの管理（Manage Policies）] をクリックします。

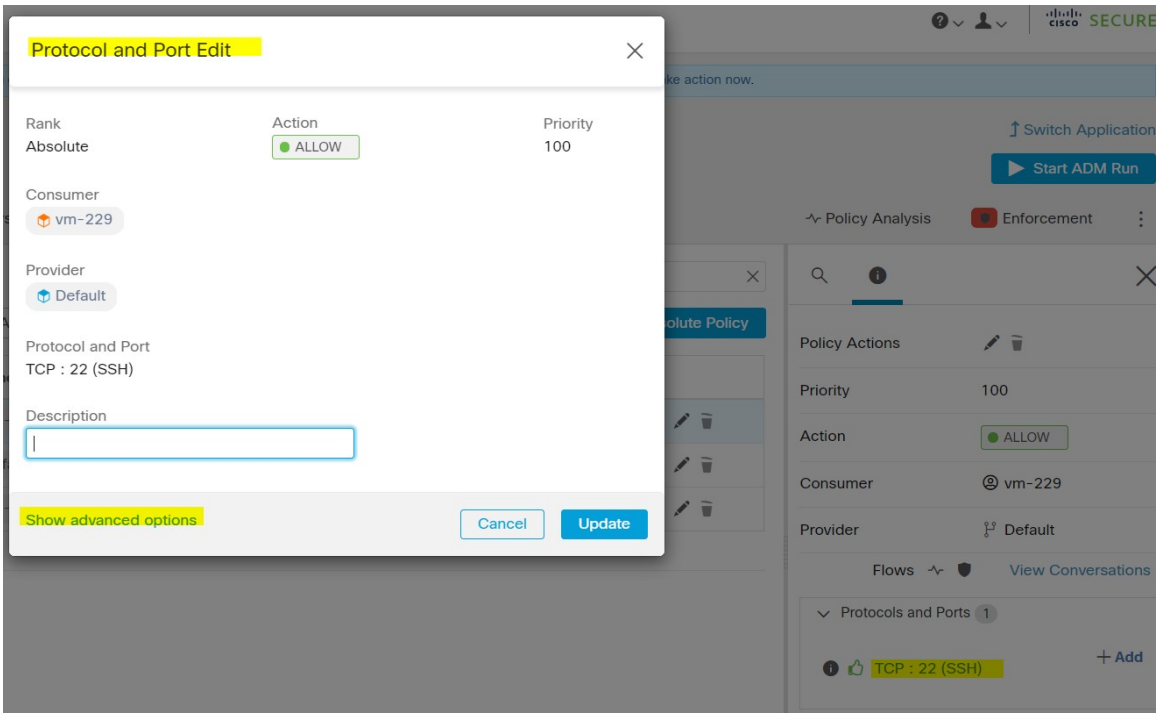
ステップ 5 編集するポリシーを選択します。

重要 コンシューマとプロバイダーには、Windows ワークロードのみを含める必要があります。

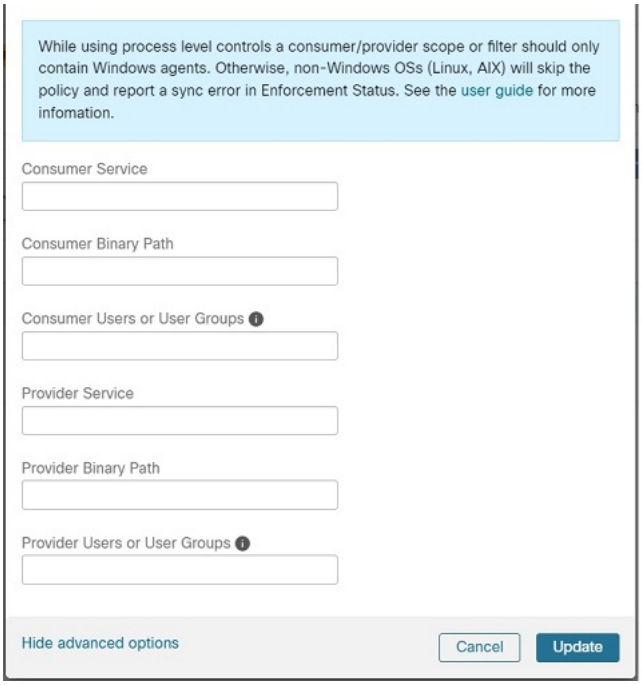
ステップ 6 編集するポリシーのテーブル行で、[プロトコルとポート（Protocols and Ports）] 列の既存の値をクリックします。

ステップ 7 右側のペインで、[プロトコルとポート（Protocols and Ports）] の下にある既存の値をクリックします。

この例では、[TCP : 22 (SSH)] をクリックします。



ステップ 8 [詳細オプションの表示（Show advanced options）] をクリックします。



ステップ 9 アプリケーション名、サービス名、またはユーザー名に基づいてコンシューマフィルタを設定します。

- アプリケーション名はフルパス名にする必要があります。

- サービス名は短いサービス名にする必要があります。
- ユーザー名は、ローカルユーザー名（tetter など）またはドメインユーザー名（sensor-dev@sensor-dev.com、sensor-dev\sensor-dev など）にできます。
- ユーザーグループは、ローカルユーザーグループ（管理者など）またはドメインユーザーグループ（domain users\\sensor-dev など）にできます。
- 複数のユーザー名および（または）ユーザーグループ名は「,」で区切って指定できます（例：sensor-dev\@sensor-dev.com,domain users\\sensor-dev）。
- サービス名とユーザー名は同時に設定できません。

ステップ 10 アプリケーション名、サービス名、またはユーザー名に基づいてプロバイダーフィルタを設定します。

前のステップのコンシューマフィルタと同じガイドラインに従います。

ステップ 11 必要に応じてバイナリへのパスを入力します。

たとえば、**`c:\test\putty.exe`** と入力します。

ステップ 12 [更新（Update）]をクリックします。

既知の制限事項

- Windows 2008 R2 は、Windows OS ベースのフィルタリングポリシーをサポートしていません。
- ネットワークポリシーは単一のユーザー名で設定できますが、MS Firewall UI は複数のユーザーをサポートします。

警告

- Windows OS ベースのポリシーを使用している場合、コンシューマやプロバイダーの範囲またはフィルタにはWindowsエージェントのみを含める必要があります。そうしないと、Windows 以外の OS（Linux、AIX）ではポリシーがスキップされ、適用ステータスで同期エラーが報告されます。
- フィルタリング基準が緩い Windows OS フィルタを作成しないでください。不要なネットワークポートが開く可能性があります。
- ネットワークフローのプロセスコンテキスト、ユーザーコンテキスト、またはサービスコンテキストに関する知識が乏しいか、知識がまったくないために、ポリシーに Windows OS ベースのフィルタがある場合、ポリシー分析に矛盾が生じます。

Windows OS ベースのフィルタリング属性を使用したポリシーの確認とトラブルシューティング

Windows OS ベースのフィルタリング属性を使用する場合は、次のトピックを使用して、ポリシーが期待どおり動作することをワークロードで確認します。

Cisco TAC は、必要に応じてこの情報を使用して、このようなポリシーのトラブルシューティングを行うことができます。

アプリケーション名に基づくポリシー

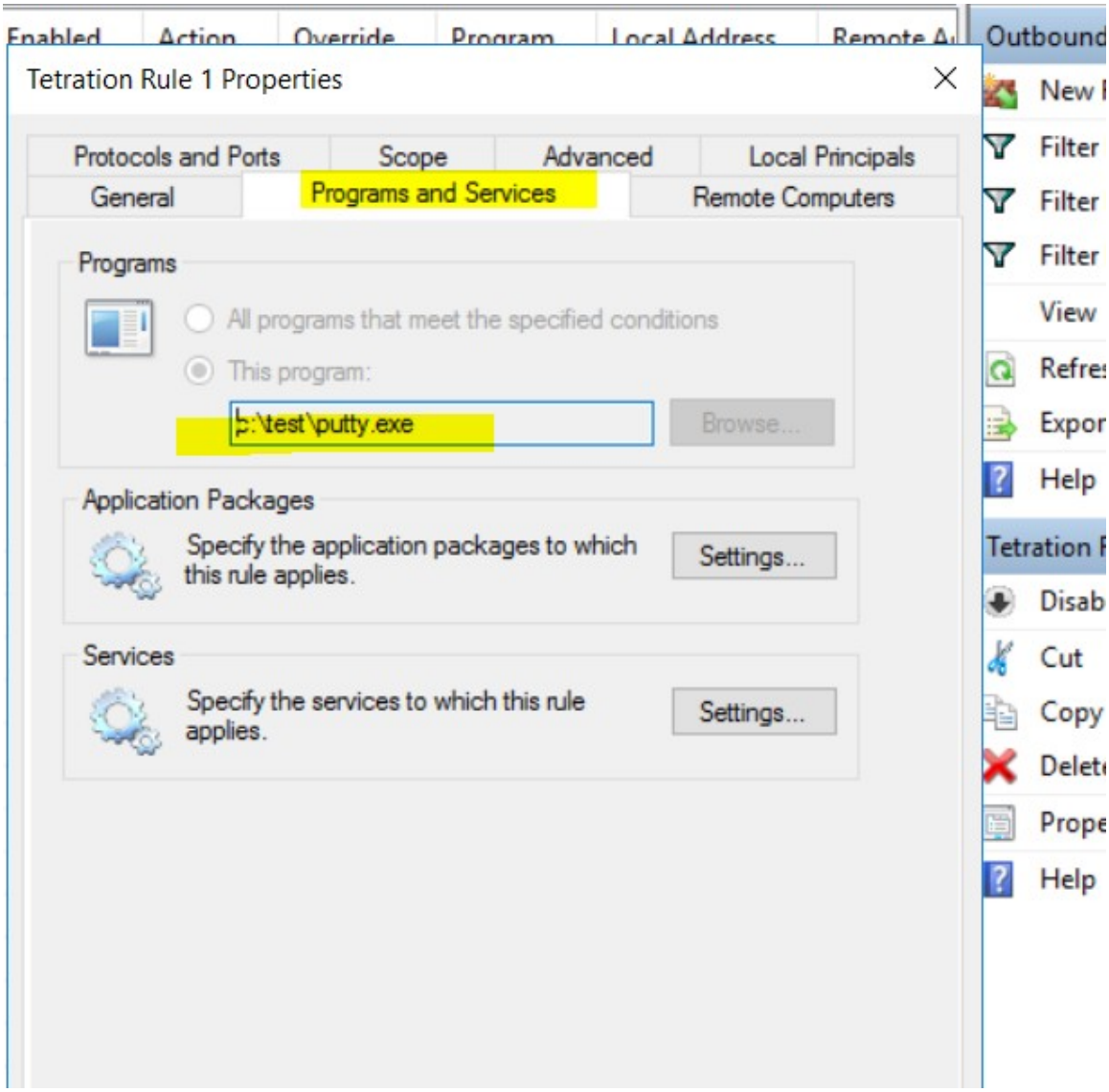
次の情報を使用して、アプリケーション名に基づく Windows OS ワークロードのポリシーを確認およびトラブルシューティングします。

次のセクションでは、**c:\test\putty.exe** として入力されたアプリケーションバイナリのワークロードにポリシーを表示させる方法について説明します。

アプリケーション名に基づくポリシーの例

```
dst_ports {
start_port: 22
end_port: 22
consumer_filters {
application_name: “c:\test\putty.exe”
}
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

生成されたファイアウォールルール



netsh を使用して生成されたフィルタ

高度なポリシーにフィルタが追加されていることをネイティブの Windows ツールで確認するには、次の手順を実行します。

- 「管理者」権限を使用して「cmd.exe」を実行します
- 「netsh wfp show filters」を実行します
- 出力ファイル filters.xml が、現在のディレクトリに生成されます。
- 出力ファイル（filters.xml）の FWPM_CONDITION_ALE_APP_ID でアプリケーション名を確認します。

```
<fieldKey>FWPM_CONDITION_ALE_APP_ID</fieldKey>
                      <matchType>FWP_MATCH_EQUAL</matchType>
                      <conditionValue>
                             <type>FWP_BYTE_BLOB_TYPE</type>
                             <byteBlob>
                                    <data>
‹→5c00640065007600690063006500 5c006800610072006400640069007300 6b0076006f006
‹→</data>
                                 <asString>\device\harddiskvolume2\temp\putty.exe</
‹→asString>
                 </byteBlob>
        </conditionValue>
```

tetenf.exe -l -f を使用して生成された WFP フィルタ

```
Filter Name:                    Secure Workload Rule 1
---------------------------------------------------------
EffectiveWeight:                18446744073709551592
LayerKey:                       FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                         Permit
RemoteIP:                       10.195.210.15-10.195.210.15
Remote Port:                    22
Protocol:                       6
AppID:                          \device\harddiskvolume2\test\putty.exe
```

アプリケーション名が無効な場合

- WAF モードでは、無効なアプリケーション名に対してファイアウォールルールが作成されます。
- WFP モードでは、無効なアプリケーション名に対して WFP フィルタは作成されませんが、NPC は拒否されません。エージェントは警告メッセージをログに記録し、残りのポリシールールを構成します。

サービス名に基づくポリシー

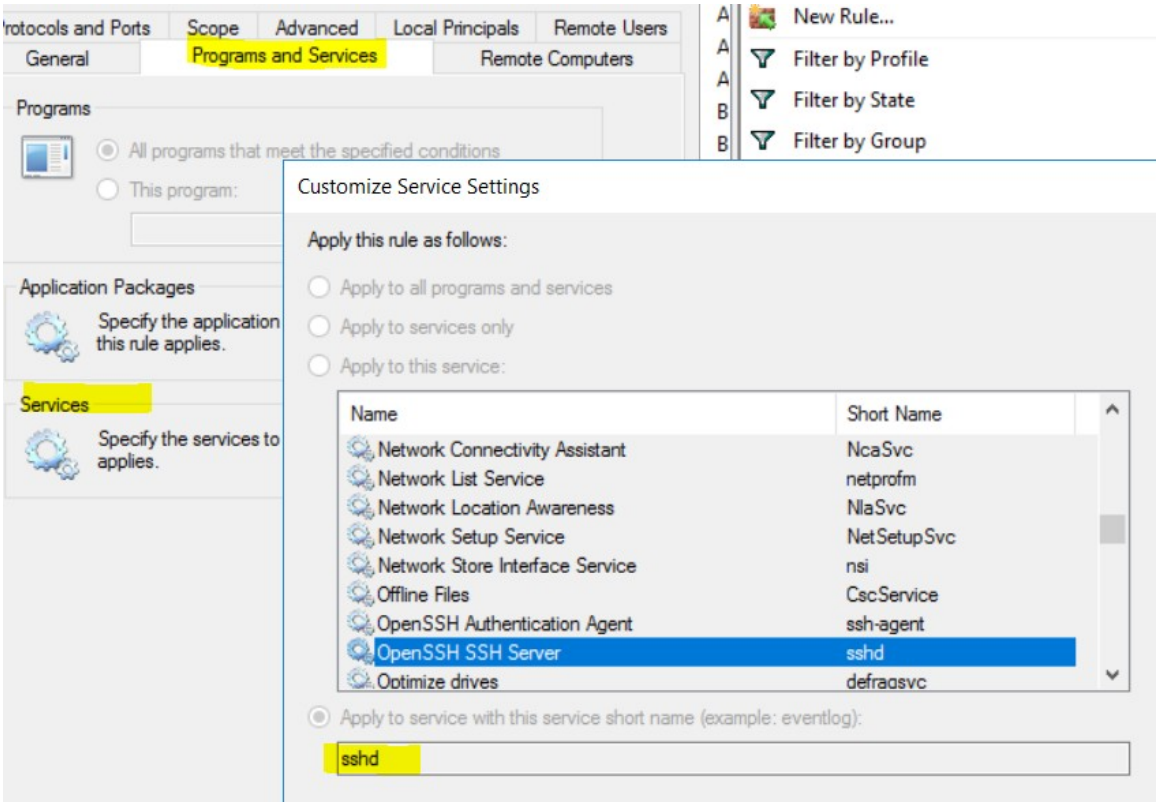
次の情報を使用して、サービス名に基づく Windows OS ワークロードのポリシーを確認およびトラブルシューティングします。

次のセクションでは、ワークロードにポリシーを表示させる方法について説明します。

サービス名に基づくサンプルポリシー

```
    dst_ports {
             start_port: 22
             end_port: 22
             provider_filters {
                    service_name: “sshd”
             }
         }}
         ip_protocol: TCP
         address_family: IPv4
         inspection_point: INGRESS
```

生成されたファイアウォールルール



netsh を使用して生成されたフィルタ

高度なポリシーにフィルタが追加されていることをネイティブの Windows ツールで確認するには、次の手順を実行します。

- 「管理者」権限を使用して「cmd.exe」を実行します
- 「netsh wfp show filters」を実行します
- 出力ファイル filters.xml が現在のディレクトリに生成されます
- 出力ファイル（filters.xml）でユーザー名の FWPM_CONDITION_ALE_USER_ID を確認します。

```
<item>
                    <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
                    <matchType>FWP_MATCH_EQUAL</matchType>
                    <conditionValue>
                              <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>

<sd>O:SYG:SYD:(A;;CCRC;;;S-1-5-80-3847866527-469524349-687026318-
→516638107)</sd>
                    </conditionValue>
</item>
```

tetenf.exe -l -f を使用して生成された WFP フィルタ

```
Filter Name:       Secure Workload Rule 3
-----------------------------------------------------
EffectiveWeight:            18446744073709551590
LayerKey:                   FWPM_LAYER_ALE_AUTH_RECV_ACCEPT_V4
Action:                     Permit
Local Port:                 22
Protocol:                   6
User or Service:            NT SERVICE\sshd
```

サービス名が不正な場合

- WAF モードで、存在しないサービス名に対してファイアウォールルールが作成されます
- WFP モードでは、存在しないサービス名に対して WFP フィルタは作成されません
- サービス SID タイプは「無制限」または「制限付き」である必要があります。サービスの種類が「なし」の場合、ファイアウォールルールと WFP フィルタを追加できますが、効果はありません。

  SID タイプを確認するには、次のコマンドを実行します。

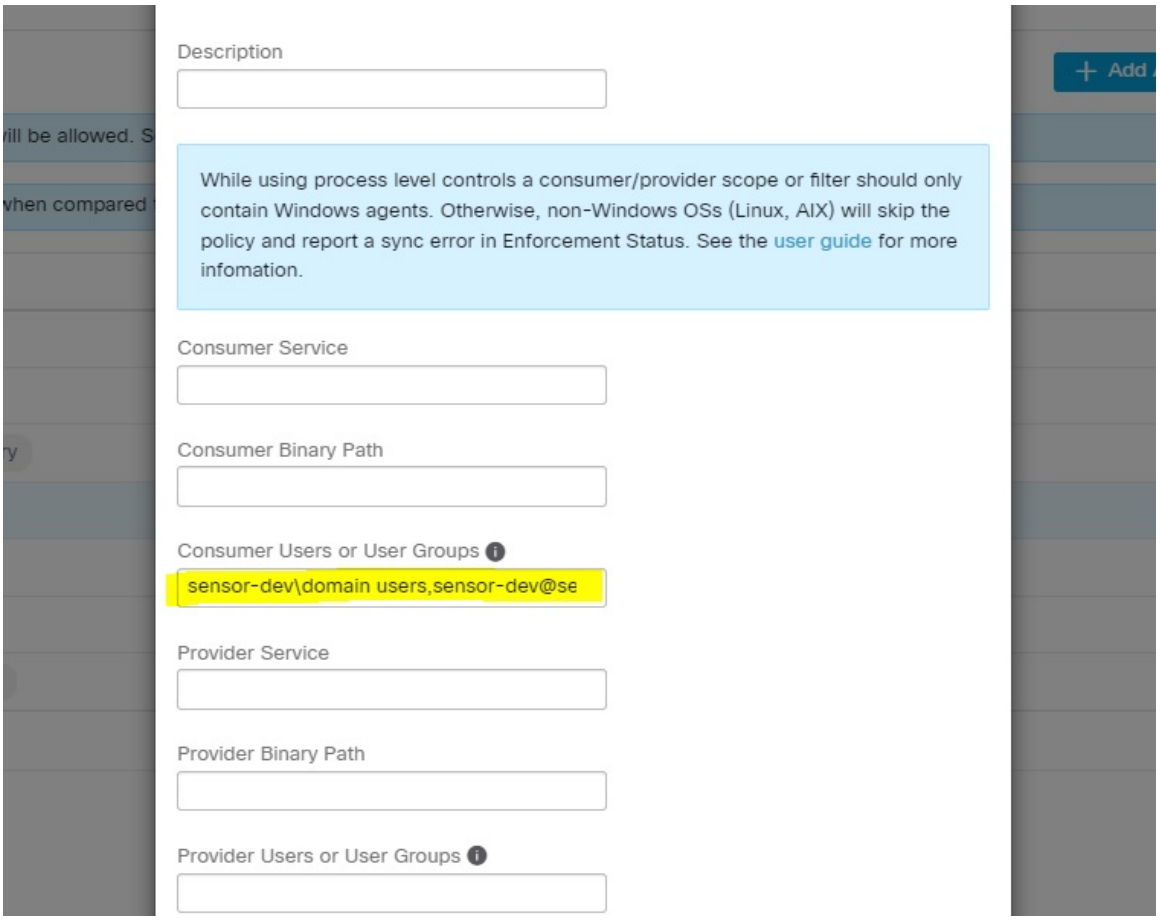
  ```
  sc qsidtype <service name>
  ```

ユーザーグループまたはユーザー名に基づくポリシー

次の情報を使用して、ユーザーグループ名の有無にかかわらず、ユーザー名に基づく Windows OS ワークロードのポリシーを確認およびトラブルシューティングします。

このトピックのセクションでは、ワークロードにポリシーを表示する方法について説明します。

このトピックの例は、次の情報を使用して設定されたポリシーに基づいています。



ユーザー名に基づくサンプルポリシー

```
dst_ports {
          start_port: 30000
          end_port: 30000
          provider_filters {
               user_name: “sensor-dev\sensor-dev”
           }
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

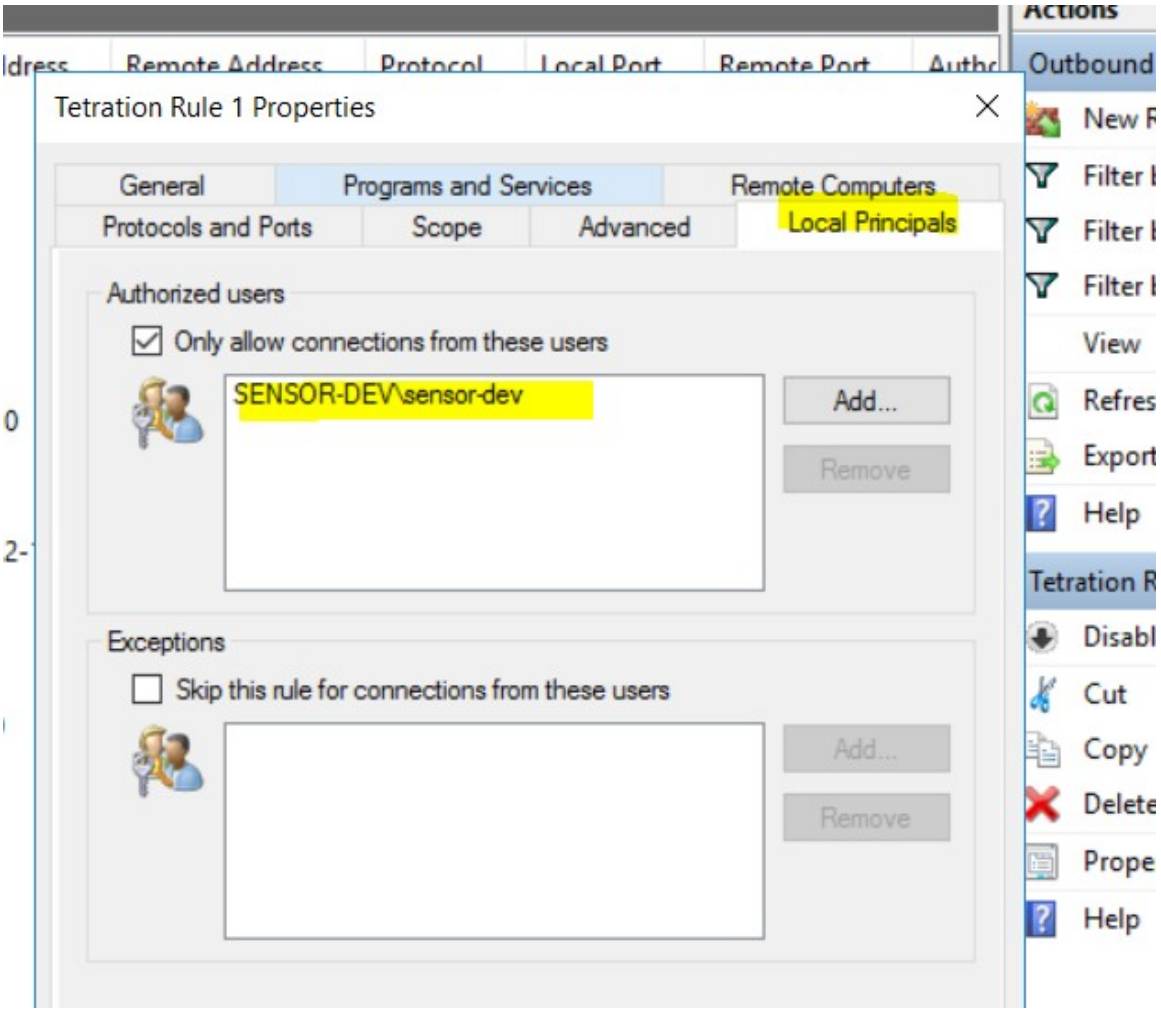
ユーザーグループとユーザー名に基づくサンプルポリシー

```
dst_ports {
start_port: 30000
end_port: 30000
provider_filters {
user_name: “sensor-dev\domain users,sensor-dev\sensor-dev”
}
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

生成されたファイアウォールルール

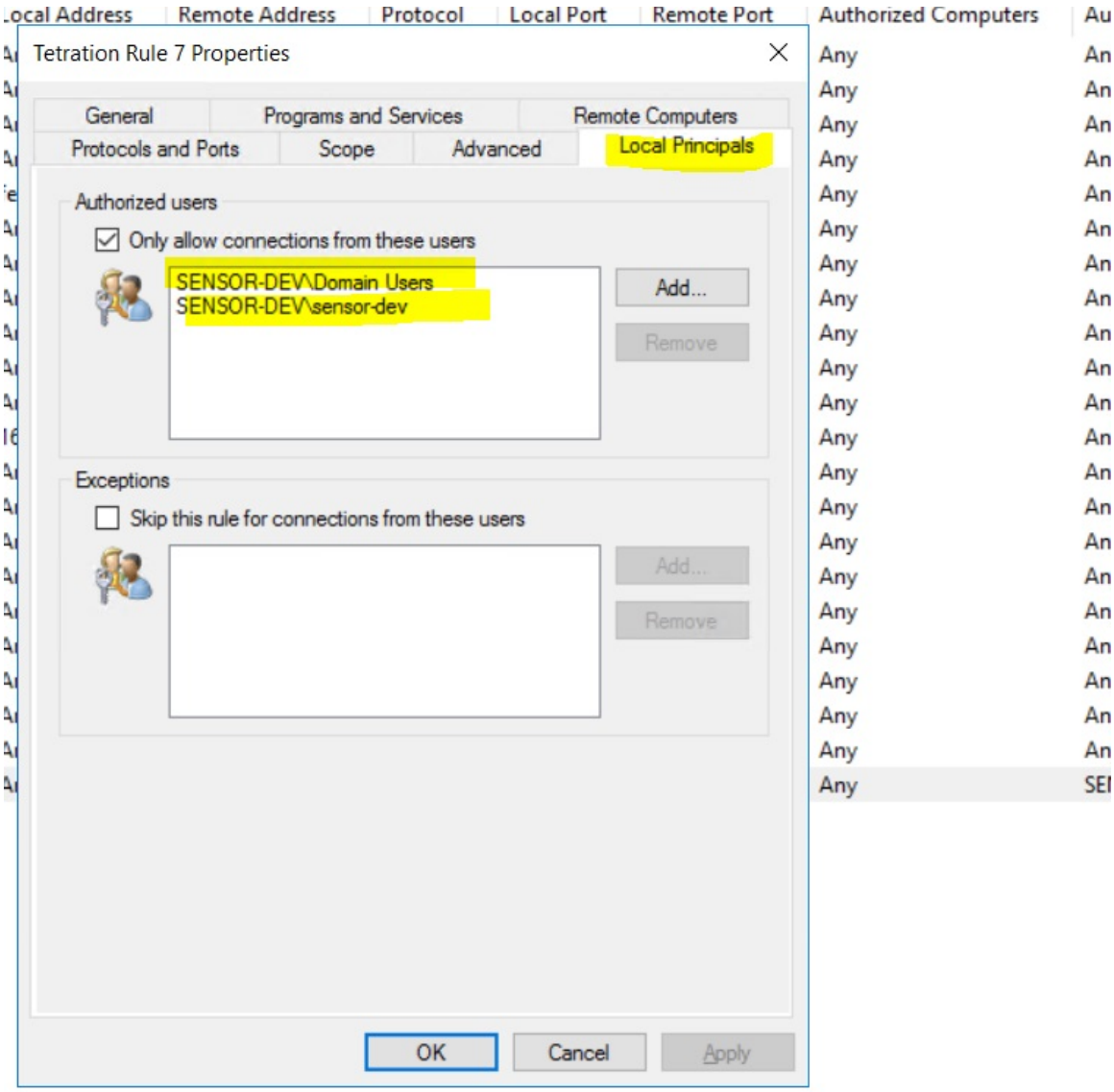
ユーザー名に基づくファイアウォールルール

例：ユーザー名 sensor-dev\\sensor-dev に基づくファイアウォールルール



ユーザーグループとユーザー名に基づくファイアウォールルール

例：ユーザー名 sensor-dev\\sensor-dev およびユーザーグループ domain users\\sensor-dev に基づくファイアウォールルール



netsh を使用して生成されたフィルタ

高度なポリシーにフィルタが追加されていることをネイティブの Windows ツールで確認するには、次の手順を実行します。

- 「管理者」権限を使用して「cmd.exe」を実行します
- 「netsh wfp show filters」を実行します
- 出力ファイル filters.xml が、現在のディレクトリに生成されます。
- 出力ファイル（filters.xml）でユーザー名の FWPM_CONDITION_ALE_USER_ID を確認します。

```
<item>
            <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
            <matchType>FWP_MATCH_EQUAL</matchType>
            <conditionValue>
                   <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
                  <sd>O:LSD:(A;;CC;;;S-1-5-21-4172447896-825920244-2358685150)</sd>

            </conditionValue>
</item>
```

tetenf.exe -l -f を使用して生成された WFP フィルタ

ユーザー名に基づくフィルタ

例：ユーザー名 SENSOR-DEV\sensor-dev に基づく WFP ルール

```
Filter Name:                    Secure Workload Rule 1
---------------------------------------------------------
EffectiveWeight:                18446744073709551590
LayerKey:                       FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                         Permit
RemoteIP:                       10.195.210.15-10.195.210.15
Remote Port:                    30000
Protocol:                       6
User or Service:                SENSOR-DEV\sensor-dev
```

ユーザーグループとユーザー名に基づくフィルタ

例：ユーザー名 SENSOR-DEV\\sensor-dev およびユーザーグループ名 SENSOR-DEV\\Domain Users に基づく WFP ルール

```
Filter Name:          Secure Workload Rule 1
-----------------------------------------------------
EffectiveWeight:              18446744073709551590
LayerKey:                     FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                       Permit
RemoteIP:                     10.195.210.15-10.195.210.15
Remote Port:                  30000
Protocol:                     6
User or Service:              SENSOR-DEV\Domain Users, SENSOR-DEV\sensor-dev
```

ネットワークポリシールールにサービス名とユーザー名を設定することはできません。

ユーザー名またはユーザーグループが不正な場合

- ユーザー名またはユーザーグループが無効な場合、ネットワークポリシーはWindowsエージェントによって拒否されます。

Kubernetes と OpenShift

（オプション）Kubernetes ワークロードの追加ポリシー

次の手順は、Kubernetes 環境に応じたオプションです。

ホストネットワークモードで実行されている *Kubernetes Nginx Ingress* コントローラのポリシー

Cisco Secure Workload は、Kubernetes Ingress オブジェクトを使用して外部クライアントにポッドが公開されるときに、Nginx Ingress コントローラとバックエンドポッドの両方でポリシーを適用します。

（注） Ingress コントローラがホストネットワークモードで実行されていない場合は、IngressControllerAPI を参照してください。

（注） IBM-ICP は、デフォルトで Kubernetes Nginx Ingress コントローラを使用し、ホストネットワークモードのコントロールプレーンノードで実行されます。

Kubernetes Nginx Ingress コントローラを使用してポリシーを適用する手順は次のとおりです。

手順

ステップ 1 ここに記載する説明に従って、Kubernetes または OpenShift の外部オーケストレータを作成します。

```
→  ~
→  ~ k8s get ingress
NAME           HOSTS   ADDRESS          PORTS   AGE
test-ingress   *       192.168.60.100   80      7s
→  ~
```

ステップ 2 Kubernetes クラスタに入力オブジェクトを作成します。入力オブジェクトの作成で使用する yaml ファイルのスナップショットを次の図に示します。

```
▶ k8s get ingress
NAME                       HOSTS   ADDRESS          PORTS   AGE
svc-ce2e-teeksitlbiwlc     *       192.168.10.13    80      74s
```

```
~
▶ k8s get ingress -o yaml
apiVersion: v1
items:
- apiVersion: extensions/v1beta1
  kind: Ingress
  metadata:
    annotations:
      virtual-server.f5.com/ip: 192.168.10.13
      virtual-server.f5.com/partition: k8scluster
    creationTimestamp: "2020-06-26T21:31:01Z"
    generation: 1
    labels:
      e2e-test: "yes"
    name: svc-ce2e-teeksitlbiwlc
    namespace: default
    resourceVersion: "1074475"
    selfLink: /apis/extensions/v1beta1/namespaces/default/ingresses/svc-ce2e-teeksitlbiwlc
    uid: 5526b4a3-b7f4-11ea-aa09-525400d58002
  spec:
    backend:
      serviceName: svc-ce2e-teeksitlbiwlc
      servicePort: 80
  status:
    loadBalancer:
      ingress:
      - ip: 192.168.10.13
kind: List
metadata:
  resourceVersion: ""
  selfLink: ""
```

ステップ 3 Kubernetes クラスタに Kubernets Nginx Ingress コントローラを展開します。IBM-ICP Ingress コントローラポッドは、デフォルトでコントロールプレーンノードで実行されています。

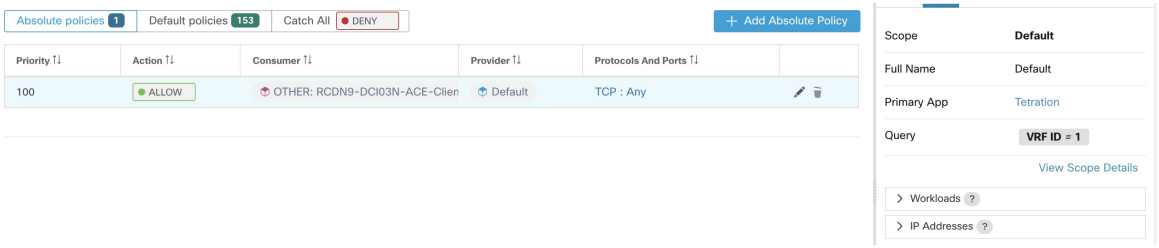
```
~
▶ k8s get pods -o wide -n ingress-nginx
NAME                                       READY   STATUS    RESTARTS   AGE     IP              NODE                        NOMINATED NODE
nginx-ingress-controller-6bc9c6745c-scfzs   1/1     Running   0          2m11s   192.168.10.13   enforcement-scale-16-kube3   <none>

~
▶ k8s get node enforcement-scale-16-kube3 -o wide
NAME                        STATUS   ROLES    AGE    VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE           KERNEL-VERSION      CONTAINER-RUNTIME
enforcement-scale-16-kube3   Ready    <none>   7d5h   v1.12.3   192.168.10.13   <none>        Ubuntu 16.04.5 LTS   4.4.0-139-generic   docker://18.6.1
```

ステップ 4 クラスタ外のコンシューマがアクセスするバックエンドサービスを作成します。次の例では、簡単な *svc-ce2e-teeksitlbiwlc* （http-echo）サービスを作成しています。

```
~
▶ k8s get svc svc-ce2e-teeksitlbiwlc
NAME                    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
svc-ce2e-teeksitlbiwlc   ClusterIP   10.102.30.231   <none>        80/TCP    6m11s
```

ステップ 5 外部コンシューマとバックエンドサービスの間にポリシーを作成します。



ステップ 6 準備ができたら、ポリシーを適用します。

ステップ 7 Nginx Ingress コントローラの場合、Secure Workload ソフトウェアでは、送信元は上記の手順で指定されたコンシューマであり、宛先は対応する Ingress コントローラポッド IP である、適切な許可/ドロップルールが適用されます。バックエンドポッドの場合、Secure Workload ソフトウェアでは、送信元は入力ポッドであり、宛先はバックエンドポッド IP である、適切な許可/ドロップルールが適用されます。

Deployment/Daemonset として実行されている *Kubernetes Nginx/Haproxy Ingress* コントローラのポリシー

Cisco Secure Workload は、Kubernetes Ingress オブジェクトを使用して外部クライアントにポッドが公開されるときに、Ingress コントローラとバックエンドポッドの両方でポリシーを適用します。

Ingress コントローラにポリシーを適用する手順は次のとおりです。

手順

ステップ 1 OpenAPI を使用して、Kubernetes/OpenShift の外部オーケストレータを作成/更新します。OpenAPI を使用して外部オーケストレータを作成する方法については、「オーケストレーション」を参照してください。外部オーケストレータ設定のための Ingress コントローラの情報を追加します。

ステップ 2 Kubernetes クラスタで Ingress オブジェクトを作成します。

ステップ 3 Kubernetes クラスタで Ingress コントローラを展開します。

ステップ 4 クラスタ外のコンシューマがアクセスするバックエンドサービスを作成します。

ステップ 5 外部コンシューマとバックエンドサービスの間にポリシーを作成します。

ステップ 6 準備ができたら、ポリシーを適用します。

ステップ 7 Ingress コントローラの場合、Secure Workload ソフトウェアでは、送信元は上記の手順で指定されたコンシューマであり、宛先は対応する Ingress コントローラポッド IP である、適切な許可/ドロップルールが適用されます。バックエンドポッドの場合、Secure Workload ソフトウェアでは、送信元は入力ポッドであり、宛先はバックエンドポッド IP である、適切な許可/ドロップルールが適用されます。

ワークロードのグループ化：クラスタとインベントリフィルタ

クラスタとインベントリフィルタの目的は似ていますが、いくつかの重要な違いがあります。

表 *28*：クラスタとインベントリフィルタの比較

クラスタ	[インベントリフィルタ（Inventory Filters）]
範囲内のワークロードのサブセットにポリシーを適用するために使用されます。	範囲内のワークロードのサブセットにポリシーを適用するために使用できます。 また、範囲に関係なくワークロードにポリシーを適用するためにも使用できます（たとえば、特定のオペレーティングシステムを実行しているすべてのワークロードにポリシーを適用するために使用できます）。
クエリによって定義されます。	クエリによって定義されます。
単一範囲内のワークロードのみを含めることができます。	単一範囲に制限されたメンバーシップを持つことも、任意の範囲内のワークロードを含めることもできます（たとえば、フィルタがオペレーティングシステムに基づいている場合）。
同じワークスペースおよびワークスペースバージョンのポリシーでのみ使用できます。	任意の範囲と任意のワークスペースのポリシーで使用できます。
自動ポリシー検出中に自動的に作成できます。	手動で作成するか、既存のクラスタから変換する必要があります。
承認されていない場合は、自動ポリシー検出中に上書きできます。既知の良好なクラスタを承認すると、今後の検出実行で他のクラスタの精度を向上させることができます。	自動ポリシー検出によって変更されることはありません。
自動ポリシー検出の重要な機能を活用できます。次の機能があります。 • グループ内のワークロードが一緒に属しているかどうかを評価するために役立つ信頼度評価があります。 • 同じワークスペースで他のポリシー検出の実行中に生成されたクラスタと比較できます。	--

クラスタ	[インベントリフィルタ（Inventory Filters）]
外部依存関係 （549ページ）を設定する場合、およびクロス範囲ポリシーとポリシー検出に関連するその他の機能を設定する場合は使用できません。	外部依存関係、およびクロス範囲ポリシーに関連するその他の機能（オートパイロットルールなど）を含む、きめ細かいポリシーを設定するために使用できます。
「クラスタ （596ページ）」およびサブトピックを参照してください。	インベントリフィルタの作成（453ページ）およびクラスタをインベントリフィルタに変換する （601 ページ）を参照してください。

クラスタ

クラスタは、ワークスペース内でグループ化された一連のワークロードです。（Secure Workloadの展開もクラスタと呼ばれることもありますが、この 2 つの用途は無関係です）

たとえば、アプリケーションの範囲に、アプリケーションを構成する他の多くのタイプのサーバーとホストの中に数台の Web サーバーが含まれている場合、このアプリケーションの範囲内に Web サーバーのクラスタが必要になる可能性があるため、これらの Web サーバーにのみ特定のポリシーを割り当てられます。

自動ポリシー検出は、設定の実行中に指定された時間枠で観測されたシグナルに基づいて、ワークロードをクラスタにグループ化します。

各クラスタはクエリによって定義される

クラスタクエリは、特定のIPアドレスで定義しない限り動的です。動的クエリを使用すると、クラスタメンバーシップは時間の経過とともに変化し、インベントリの変更を反映できます。クエリに一致させるワークロードは多いか、少ないか、または別のものになります。

たとえば、クラスタクエリが部分文字列「HR」を含むホスト名に基づいており、HR を含むホスト名を持つホストがワークスペースに追加された場合、クラスタには自動的に追加のホストが含まれます。

自動ポリシー検出は、ワークロードに関連付けられたホスト名とラベルを調べます。自動ポリシー検出により、ホスト名とこれらのラベルに基づいて候補クエリの短いリストがクラスタごとに生成されます。これらのクエリから 1 つを選択して、必要に応じて編集し、クラスタに関連付けられます。自動ポリシー検出においてホスト名とラベルに基づく簡単なクエリが作成できない場合もあり、そのときは（代替の）クエリが提案されないことに注意してください。

承認済みクラスタのワークロードは、将来のポリシー検出の影響を受けない

関連するワークスペース内で承認済みクラスタのメンバーになっていないワークロードのみが、ポリシー検出の影響を受けます。**承認済みクラスタ**は、手動で承認したクラスタです。詳細については、クラスタの承認 （605 ページ）を参照してください。

クラスタを編集してグループ化を強化する

次のセクションでは、クラスタリング結果を編集、強化、および承認するためのいくつかのワークフローについて説明します。ワークスペースの最新バージョンでのみクラスタを変更/承認できることに注意してください（「アクティビティログとバージョン履歴」を参照）。

クラスタに変更を加える （599 ページ）を参照してください。

Kubermentes インベントリを含むクラスタ

（注） ワークスペースに複数の Kubernetes 名前空間からのインベントリが含まれている場合、各クラスタクエリを名前空間でフィルタ処理する必要があります。名前空間フィルタがまだ存在しない場合は、各クエリに名前空間フィルタを追加します。クエリを変更すると、ポリシーが自動的に再検出されます。

クラスタは、単一のワークロードで構成されている場合があります。

単一のワークロードのみを含むポリシーの作成が必要になる場合があります。

クラスタはインベントリフィルタに変換される場合があります。

承認済みクラスタと同様に、インベントリフィルタに昇格されたクラスタは、その後のポリシー検出中に変更されません。

クラスタとは異なり、インベントリフィルタはワークスペースに関連付けられていませんが、Cisco Secure Workload 展開においてグローバルに使用できます。

クラスタとインベントリフィルタの比較については、ワークロードのグループ化：クラスタとインベントリフィルタ （595 ページ）を参照してください。

クラスタをインベントリフィルタに変換する （601 ページ） を参照してください。

クラスタの信頼度

クラスタの信頼度スコアまたは品質スコアを使用して、改善が必要なクラスタを特定します。

クラスタの信頼度は、メンバーワークロードの信頼度の平均値です。一般に、ワークロードが割り当てられたクラスタの他のメンバーと類似しているほど、また最も近い（最も類似した）代替クラスタのワークロードと類似していないほど、そのワークロードの信頼度は高くなります。

フローがクラスタリングに使用される場合、2つのワークロードは、カンバセーションのパターンが類似している場合は類似しています（カンバセーショングラフ内の類似したネイバーセット、つまりコンシューマワークロードおよびプロバイダーワークロードとポートの類似したセットなど）。

（注）

- 次の場合、クラスタの信頼度は計算されません（未定義になります）。
  - 1 つのワークロードのみを含むクラスタ
  - 承認済みのクラスタ
  - 通信が観測されなかった範囲内のワークロード（プロセスベースのクラスタリングが選択された場合、プロセス情報の利用は不可）
- クラスタがパーティション境界を越えて構成されることはありません（サブネット境界など。高度な自動ポリシー検出設定のルートラベルを参照）。ただし、信頼度と代替クラスタの計算では、そのような境界は無視されます。これは、異なるサブネットにあるにもかかわらず、非常によく似た動作をするワークロードまたはクラスタが存在する可能性があることを示しています。
- クラスタの編集後、ポリシーが再度検出されるまで再計算は行われないため、信頼度スコアが不正確になる可能性があります。

クラスタの信頼度を表示する方法については、「クラスタの表示（598 ページ）」を参照してください。

クラスタの表示

クラスタビューは、クエリとクラスタの関連付け、およびクエリの編集をサポートします。

クラスタビューでは、テーブルの列見出しをクリックして、その列（名前、ワークロードの数、信頼度など）に基づいてクラスタを並べ替えることができます。

各クラスタの行をクリックすると、説明、提案または承認されたクエリ、メンバーワークロードなどの詳細なクラスタ情報が右側のパネルに表示されます。これらのフィールドのいくつかは編集可能です。

クラスタとその詳細を表示するには、次の手順を実行します。

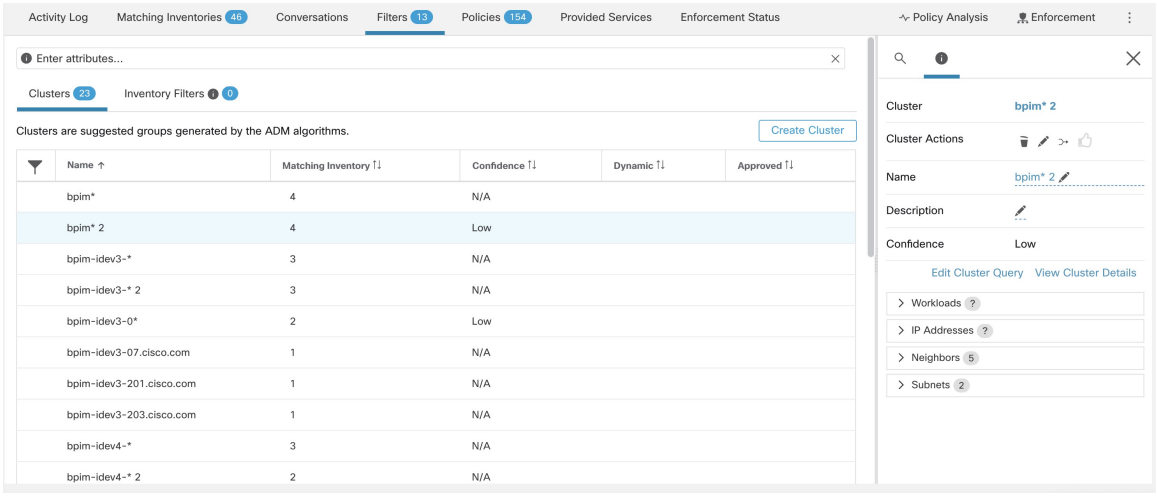
1. 目的の範囲とワークスペースに移動します。

   クラスタはワークスペースに固有です。範囲内の各ワークスペースは、異なるクラスタを持つことができます。現在のワークスペースの外部でクラスタを使用できるようにするには、クラスタをインベントリフィルタに変換する（601 ページ）を参照してください。

2. [ポリシーの管理（Manage Policies）] をクリックします。
3. [フィルタ（Filters）] をクリックします。
4. [クラスタ(Clusters)] をクリックします。
5. クラスタに関する情報を表示するには、クラスタをクリックします。
   1. 右側に開くパネルを確認します。
   2. 詳細については、[クラスタの詳細の表示（View cluster details）] をクリックします。

[クラスタの詳細（Cluster Details）] ページが別のブラウザタブで開きます。

図 *288 :* クラスタ ビュー



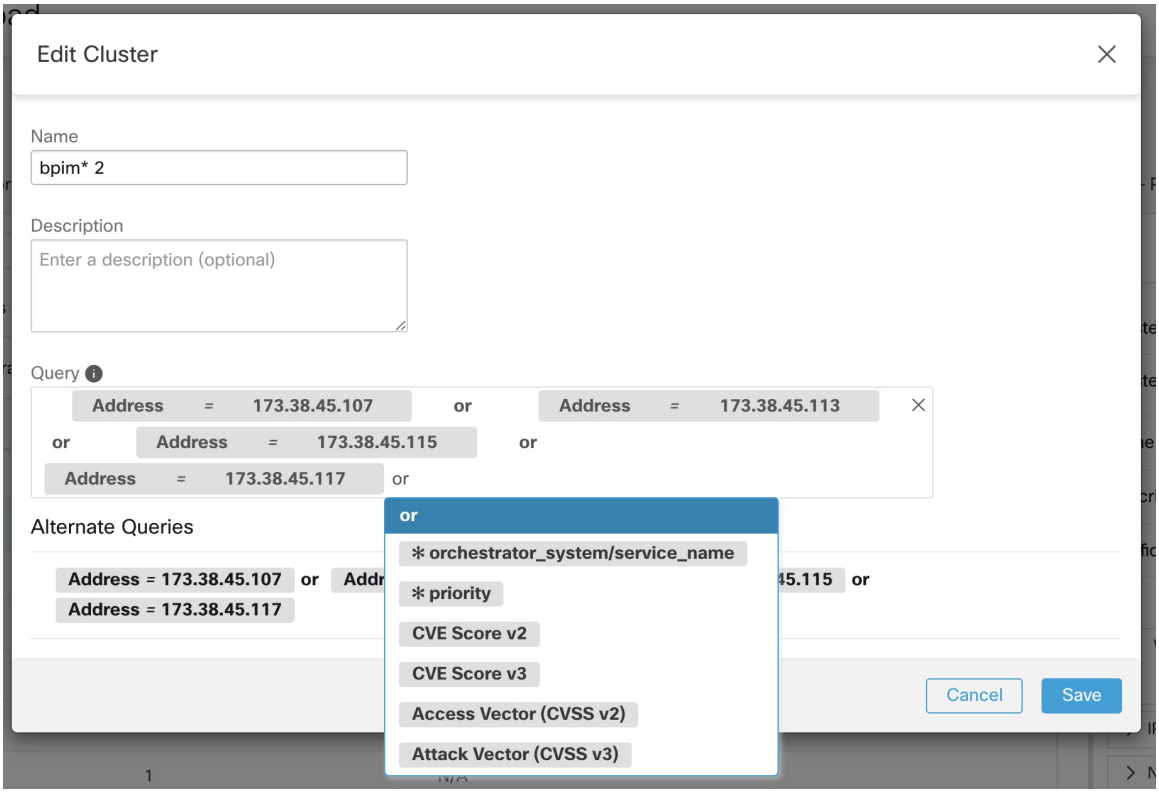
クラスタに変更を加える

自動ポリシー検出では、クラスタごとに 1 つ以上の候補クエリが作成されます。

クラスタリングの結果が期待と完全に一致しない場合は、クエリを編集してグループ化を改善できます。

クラスタを参照および編集するには：ページの上部にある [クラスタ（clusters）] ボックスをクリックします。クラスタを変更、たとえば、クラスタのメンバーを変更するか、そのクエリを選択または変更するには、以下に示すように、クラスタのクエリを選択または編集します。

図 *289* : クラスタの編集



明示的な IP アドレスを追加または削除するか、提供された代替のリストから別のクエリを選択して、クエリを編集できます。クラスタのクエリは、アドレス、ホスト名、およびラベルで表現されたクエリフィルタにできます。明示的な IP アドレスではなくラベルに基づいてクエリを定義すると、クラスタは動的になり、適切にラベル付けされた新規、変更、または削除されたインベントリは、クラスタに自動的に含まれるか、クラスタから除外されます。

クエリの選択と可能な編集が完了したら、[保存（Save）] をクリックします。[保存（Save）] ボタンをクリックすると、クラスタは自動的に承認済みとしてマークされ、承認済みの親指アイコンが（変更の有無に関係なく）青色に変わります。必要に応じて、承認済みアイコンを切り替えて、承認済みステータスを変更できます。詳細については、クラスタの承認（605 ページ）を参照してください。



重要 クラスタのメンバーシップが変更された場合、変更されたクラスタ間のフローの変更が正確に反映されている更新されたポリシーを取得するために、ポリシーを再度検出する必要がある場合があります。これは、クラスタへの新しいノードの追加などにより、クラスタメンバーシップが変更された可能性があるためです。ワークスペースに対応する範囲が編集された場合、または一般にワークスペースのメンバーシップが変更された場合、同様の状況が発生する可能性があります。同様に、クラスタのメンバーシップが変更されると、クラスタの確実性スコアが正確でなくなる可能性があります。これらすべての場合において、ポリシーの自動検出は、更新されたポリシーとクラスタの確実性スコア（未承認のクラスタの更新された確実性）を取得するのに役立ちます。

クラスタクエリを編集すると、そのクエリに関連付けられたクラスタが重複する可能性があります。

クラスタをインベントリフィルタに変換する

次の場合、クラスタをインベントリフィルタに変換します。

- クラスタを承認するためのより用途の広い代替手段として、将来の自動ポリシー検出の実行によってクラスタが変更されないようにする必要があります。
- クラスタをワークスペースとワークスペースのバージョンから独立させる必要があります。
- コンシューマとプロバイダーが異なる範囲に属するポリシーを作成または検出し、範囲全体に関係するポリシーだけでなく、範囲内のワークロードのサブセットに固有のポリシーを作成する必要があります。

  コンシューマとプロバイダーが異なる範囲にある場合：ポリシーオプション（615ページ）で説明されている詳細な方法を使用してクロス範囲ポリシーを作成する場合や、範囲ごとよりも細かいポリシーが必要な場合は、クラスタの代わりにインベントリフィルタを使用する必要があります。

手順

ステップ 1 昇格するクラスタを含むワークスペースに移動します。

ステップ 2 [ポリシーの管理（Manage Policies）] をクリックします。

ステップ 3 [フィルタ（Filters）] をクリックします。

ステップ 4 [クラスタ(Clusters)] をクリックします。

ステップ 5 クロス範囲ポリシーで使用するクラスタをクリックします。

ステップ 6 右側のパネルにある [クラスタアクション（Cluster Actions）] セクションで、（[インベントリフィルタに昇格（Promote to Inventory Filter）]）をクリックします。

ステップ 7 名前、説明、およびクエリが期待どおりの内容であることを確認します。

ステップ 8 [クエリを所有権の範囲に制限する（Restrict Query to Ownership Scope）] を選択します。

（インベントリフィルタが範囲の境界を越えることはできますが、この目的のためには不要な動作です。このフィルタにはこの範囲内のワークロードのみを含める必要があります）。

ステップ 9 このインベントリフィルタによって定義されたアプリケーションを、自動ポリシー検出中に生成されたポリシーのプロバイダーにする場合は、[範囲外のサービスを提供する（Provides a service external of its scope）] を選択します。

このアプリケーションがプロバイダーではなくコンシューマである場合、または手動で作成されたポリシーに対してのみこのインベントリフィルタを使用する場合は、このオプションを有効にする必要はありません。

ステップ 10 [クラスタの昇格（Promote Cluster）] をクリックします。

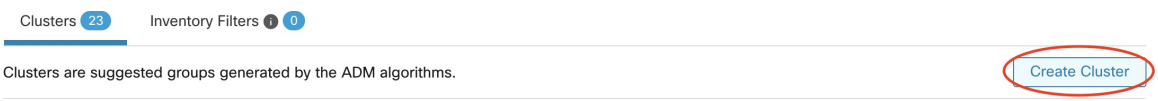
ステップ 11 クラスタが [インベントリフィルタ（Inventory Filters）] タブに移動したことを確認します。

確認するには、ページの更新が必要な場合があります。

クラスタの作成または削除

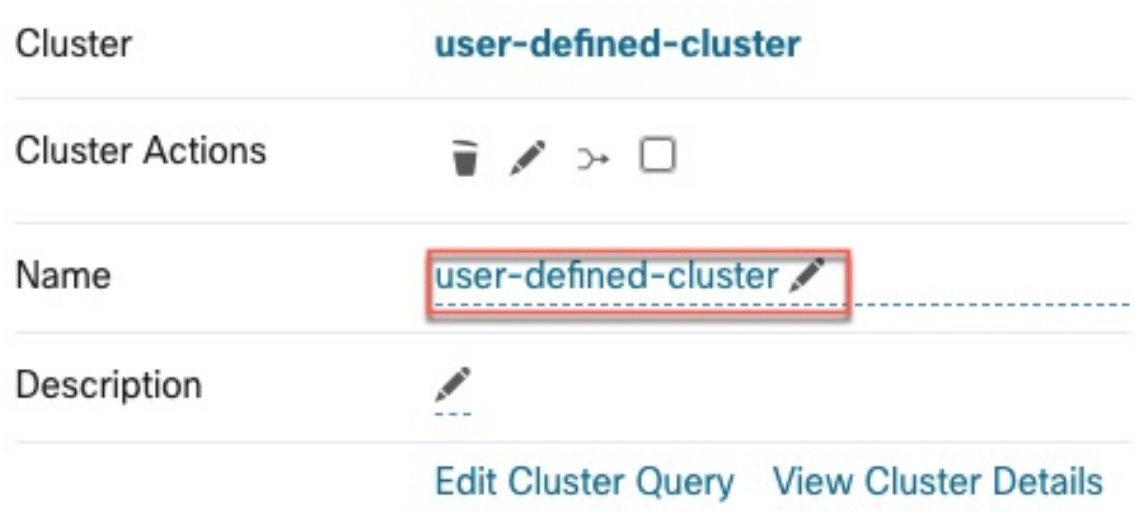
クラスタページの [クラスタの作成（Create Cluster）] ボタンをクリックして、新しい空のクラスタを作成します。または、[開始する（Get Started）] サイドバーの [フィルタの作成（Create Filter）] ボタンをクリックし、モーダルで [クラスタ（Clusters）] を選択して、[自動ポリシー検出（Automatic Policy Discovery）] ページからクラスタを作成できます。

図 *290* : 新しいクラスタの作成



新しいユーザー定義クラスタはサイドパネルに表示され、必要に応じて名前を変更できます。

図 *291* : クラスタの名前の変更



空のクラスタを削除するには、いずれかのビューでクラスタを選択して詳細をサイドパネルに表示し、クラスタ詳細ビューのヘッダーにあるごみ箱ボタンをクリックします。上の図を参照してください。

生成されたクラスタのバージョンの比較：差分ビュー

ワークスペースに対して少なくとも2回ポリシーを自動的に検出した後に、異なる検出の実行で生成されたクラスタを比較することができます。

手順

ステップ1 次のいずれかの方法を使用して、クラスタの差分ビューに移動します。

- ポリシーの検出に成功すると、検出結果を示す差分ビューに移動するリンクとともに、成功を示すメッセージが表示されます。結果のリンクをクリックします。

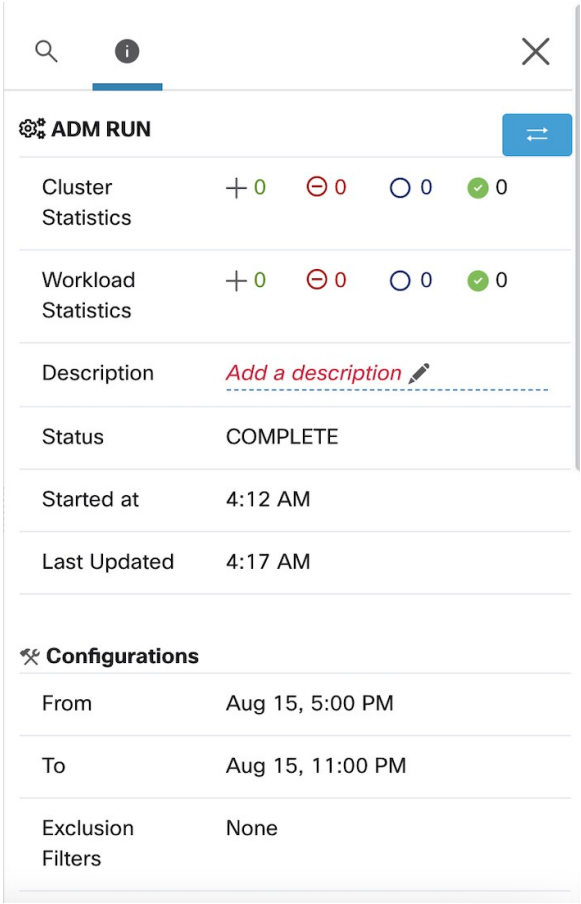
図 *292*：正常に実行された自動ポリシー検出



- バージョンビューからリビジョンを比較します。
  1. 検出されたポリシーバージョンの表示、比較、および管理 （570 ページ） の手順を実行します。
  2. [リビジョンの比較（Compare Revisions）] をクリックした後、[クラスタ（Clusters）] をクリックします。

- バージョンの詳細のサイドパネルから、次の手順を実行します。
  1. 検出されたポリシーバージョンの表示、比較、および管理 （570 ページ）のバージョンの詳細を表示するための手順に従います。
  2. サイドパネルで、自動ポリシー検出の実行のコンテキスト情報が表示されているときに、サイドパネルの右上隅にある二重矢印ボタンをクリックします。

図 *293 :* コンテキスト情報の表示



ステップ 2 比較するバージョンを選択します。

ステップ 3 比較の結果を確認します。

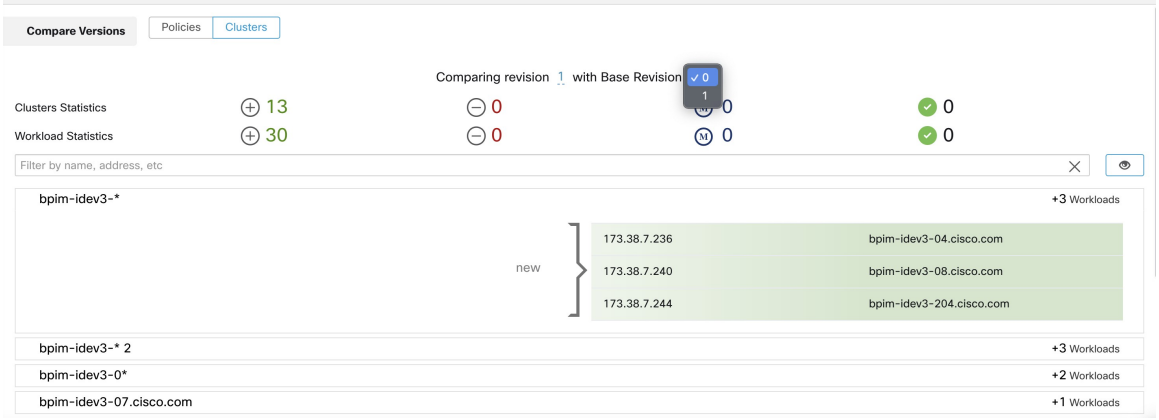
トップレベルでは、自動的に検出されたポリシーの差分ビューには、追加、削除、変更された、および未変更のクラスタとワークロードの数を示す、クラスタとワークロードの変更に関する高レベルの統計が表示されます。

ビューの残りの部分は、追加、削除、変更、未変更の順序でクラスタのリストとして編成されています。それぞれ、クラスタに追加またはクラスタから削除されたワークロードの数だけでなく、ステータスを反映するように色分けされています。

特定のクラスタやワークロードは、名前または IP アドレスで検索できます。クラスタの内容がどのように変化したかを確認するには、クラスタを表す行のいずれかをクリックして、その行を展開します。

（注） デフォルトでは、未変更のクラスタは非表示になっています。未変更のクラスタを表示するには、目のアイコンが表示されているボタンをクリックします。

図 *294* : クラスタの差分ビュー



次のタスク

ポリシーの同様の比較を表示するには、「ポリシーバージョンの比較：ポリシーの差分」を参照してください。

自動ポリシー検出の再実行中のクラスタ変更の防止

今後、ワークスペースのポリシーを自動的に検出するときに、自動ポリシー検出（旧称 ADM（によってクラスタが変更されないようにするには、クラスタを承認します。

たとえば、クラスタクエリを編集し、新しいワークロードを範囲に追加して、既存のポリシーに影響を与えずにそれらをクラスタ化する必要がある場合は、クラスタを承認します。クラスタを承認すると、クラスタのコンテンツと属性が現在の状態に固定されます。自動ポリシー検出は、承認されたクラスタを変更しません。

クラスタの承認 （605 ページ）を参照してください。

または、クラスタをインベントリフィルタに昇格させることもできます。これは、ポリシー検出によって変更されることはありません。クラスタをインベントリフィルタに変換する （601 ページ） を参照してください。

クラスタの承認

（注） クラスタをインベントリフィルタに変換する （601 ページ） も参照してください。こちらの方がより適切にニーズを満たすオプションである可能性があります。

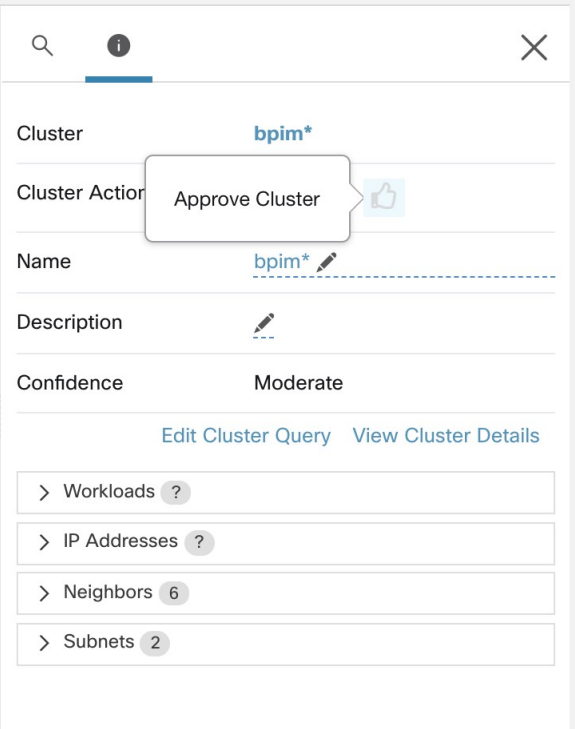
クラスタを承認しても、その後の自動ポリシー検出によってそのクラスタのクエリが変更されることはありません。承認されたクラスタのメンバーシップは、ワークスペースのメンバーが変更された場合にのみ変更される可能性があります。

承認されたクラスタのメンバーであるワークロードは、「承認されたワークロード」と呼ばれることがあります。

クラスタを承認するには、次の手順を実行します。

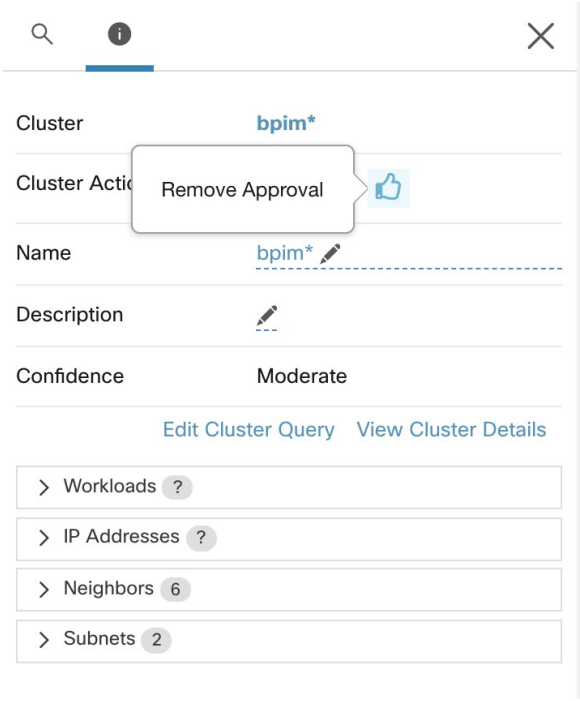
目的のクラスタがサイドパネルに表示されていることを確認します。これを行うには、クラスタを検索するか、いずれかのビューのチャートで目的のクラスタをクリックします。次に、以下に示すように、サイドパネルのクラスタ情報の右上隅にあるチェックボックスをオンにします。クラスタが承認されると、将来の自動ポリシー検出によってそのクラスタが変更されることはないことが示されます。

図 *295 :* クラスタの承認



クラスタの承認を削除するには、承認アイコンをクリックします。

図 *296* : クラスタの承認の削除



ポリシーの複雑さの対処

適用結果は、次の要因の影響を受けます。

- ルールのタイプとランク：
  - 絶対ポリシー対デフォルトポリシー
  - ワークスペースのキャッチオール設定

  ポリシーのランク：絶対、デフォルト、キャッチオール （522 ページ）を参照してください。
- ワークスペース内でのポリシーの順序

  ポリシーの優先順位 （608 ページ）を参照してください。
- キャッチオールルールを含む、親範囲または先祖範囲から継承されたポリシー

  トラフィックにヒットすることが期待されるポリシーよりも先に、優先順位の高いポリシーがそのトラフィックにヒットしないようにする必要があります。

  先祖範囲のポリシーの影響を確認するには、関連するすべての範囲でライブポリシー分析を実行します。ライブポリシー分析 （641 ページ）を参照してください。

ワークスペースでポリシーを適用する準備ができると、ワークスペースのワークロードに影響を与える継承されたポリシーがウィザードに表示されます。詳細については、ポリシー適用ウィザード （661 ページ） を参照してください。

- クロス範囲ポリシーの相互作用

  （コンシューマとプロバイダーが異なる範囲にある場合、またはカンバセーションの一方の端がポリシーとは異なる範囲にある場合）

  「コンシューマとプロバイダーが異なる範囲にある場合：ポリシーオプション （615 ページ）」を参照してください。

- ポリシー内の実際のコンシューマまたはプロバイダーが、デフォルトで設定されたコンシューマおよびプロバイダーと異なる場合がある状況（フェールオーバーシナリオなど）

  有効なコンシューマまたは有効なプロバイダー （629 ページ） を参照してください。

ポリシーの優先順位

トラフィック処理は、次の影響を受けます。

- 範囲内のポリシーの優先順位、および
- ポリシーのグローバルな順序付けと競合の解決 （608 ページ）

範囲内のポリシーの優先順位

ワークスペース内では、リスト内のポリシーの順序には各ポリシーの相対的な優先順位が反映されており、最も優先順位の高いポリシーがリストの一番上にあり、最も低い優先順位のポリシーがリストの一番下にあります。

各ワークスペースでは、絶対ポリシーがデフォルトポリシーよりも優先されます。Catch-Allポリシーはワークスペースで最も優先順位の低いポリシーです。

詳細は、ポリシーのランク：絶対、デフォルト、キャッチオール （522 ページ）を参照してください。

ポリシーのグローバルな順序付けと競合の解決

異なる範囲で定義された異なるポリシー間で競合が発生する場合があります。具体的な例を挙げると、親と子など複数の範囲に属するワークロード（インベントリ項目）に矛盾するポリシーがある場合に競合が発生します。

範囲のメンバーシップには動的な性質があるため、このような競合を手動で解決することは現実的ではありません。ワークロードは、プロパティの変更に応じて範囲に出入りできます。したがって、以下に説明するように、定義されている範囲に応じて、すべてのポリシーに対するグローバルな順位付けが必要になります。関連するポリシーのリスト（コンシューマ、プロバイダーなど範囲に応じて）がワークロードごとに識別され、グローバルな順序で並べ替えられます。フローを許可するかドロップするかは、並び替えられたリストで最初に一致したポリシーに基づいて決定されます。

ネットワーク管理者はセキュリティポリシーのグローバルな順序付けスキームを理解することで、正しい範囲とその優先順位を定義して、ワークロードに必要なポリシー全体を適用できます。アプリケーションオーナーは、各範囲内でそれぞれのワークロードにきめ細かいポリシーを適用することができます。

グローバル ネットワーク ポリシーには、次の特性があります。

- 一連の範囲が優先度順に従って（優先度の高いものから順に）並び替えられます。
- 各範囲のプライマリワークスペースには、絶対ポリシー、デフォルトポリシー、およびキャッチオールアクションが設定されています。
- 各ワークスペース内の絶対ポリシーやデフォルトポリシーの各グループは、ローカルの優先順位に従って（高いものから順に）並べ替えられます。

ポリシーのグローバルな順序は次のように定義されます。

- 全範囲のプライマリワークスペースの絶対ポリシーグループ（優先順位が高いものから順に並べられます）。
- 全範囲のプライマリワークスペースのデフォルトポリシーグループ（優先順位が低いものから順に並べられます）。
- 全範囲のキャッチオールポリシー（優先順位が低いものから順に並べられます）。

範囲の順序は、個々のポリシーではなく、カテゴリ 1 と 2 のポリシーグループに適用されることに注意してください。各グループ内では、優先順位番号が低い個々のポリシーが優先されます。

特定のワークロードの場合、最初にそれが属する範囲のサブセットが決定され、次に上記の順序が適用されます。このワークロードが属する最も優先順位の低い（適用された）ワークスペースのキャッチオールポリシーが適用可能なキャッチオールになります（ただし、絶対ポリシーやデフォルトポリシーによってオーバーライドされる場合があります）。そのワークロードの特定のフローに対して、最も一致するポリシーのアクションが適用されます。

（注）

- ワークスペースに絶対ポリシーもデフォルトポリシーも定義されていない場合、ワークスペースは無視されます。ワークスペースのキャッチオールポリシーは、グローバルな順序付けの対象ではありません。
- グローバルな順序付けにおけるデフォルトポリシーの順序は、範囲の優先順位の逆です。これにより、ポリシーの適用が有効になっていないワークスペースを含むすべてのワークスペースの境界を保護するために、すべての範囲に対して広範なポリシーを定義できます。同時に、範囲の適用を有効にしているアプリケーションオーナーは、デフォルトポリシーをオーバーライドすることができます。
- 範囲の重複は推奨されません。詳細については、範囲の重複（430ページ）を参照してください。ただし、ワークロードに2つ以上のインターフェイスがあり、範囲が重複または分離している場合、適用が有効になっている最も優先順位の低いワークスペースのキャッチオールポリシーが（適用可能なすべてのキャッチオールポリシーの中で）適用されます。

前の 3 つの範囲の例を拡張して、この順序付けスキームについて説明します。3 つの範囲に次の優先順位が割り当てられていると仮定します（範囲の優先順位を変更する方法については、「ワークスペースを使用したポリシーの管理」を参照してください）。

1. アプリ
2. アプリ：人事
3. アプリ：コマース

これらの各範囲のプライマリワークスペースには、絶対ポリシー、デフォルトポリシー、およびキャッチオールアクションが設定されています。各ワークスペース内の絶対ポリシーやデフォルトポリシーの各グループは、ローカルの優先順位に従って並べ替えられます。

ポリシーのグローバルな順序は次のとおりです。

1. アプリの絶対ポリシー
2. アプリ：人事の絶対ポリシー
3. アプリ：コマースの絶対ポリシー
4. アプリ：コマースのデフォルトポリシー
5. アプリ：人事のデフォルトポリシー
6. アプリのデフォルトポリシー
7. アプリ：コマースのキャッチオール
8. アプリ：人事のキャッチオール
9. アプリのキャッチオール

アプリの範囲に属するワークロードは、指定された順序で次のポリシーのみを受け取ります。

1. ワークロードに一致するアプリの絶対ポリシー
2. アプリのデフォルトポリシー
3. アプリのキャッチオール

アプリおよびアプリ：コマースの範囲に属するワークロードは、指定された順序で次のポリシーのみを受け取ります。

1. アプリの絶対ポリシー
2. アプリ：コマースの絶対ポリシー
3. アプリ：コマースのデフォルトポリシー
4. アプリのデフォルトポリシー
5. アプリ：コマースのキャッチオール

アプリおよびアプリ：人事の範囲に属するワークロードは、指定された順序で次のポリシーのみを受け取ります。

1. アプリの絶対ポリシー
2. アプリ：人事の絶対ポリシー
3. アプリ：人事のデフォルトポリシー
4. アプリのデフォルトポリシー
5. アプリ：人事のキャッチオール

ポリシーの順序と重複する範囲



重要 次のシナリオでは、範囲が重複しています。兄弟範囲が重複しないように注意する必要があります。ワークロードは、範囲ツリーの複数のブランチメンバーであってはなりません。詳細については、範囲の重複 （430 ページ）を参照してください。

アプリ、アプリ：人事、およびアプリ：コマースの3つの範囲に属するワークロードは、指定された順序で次のポリシーのみを受け取ります。

1. アプリの絶対ポリシー
2. アプリ：人事の絶対ポリシー
3. アプリ：コマースの絶対ポリシー
4. アプリ：コマースのデフォルトポリシー
5. アプリ：人事のデフォルトポリシー

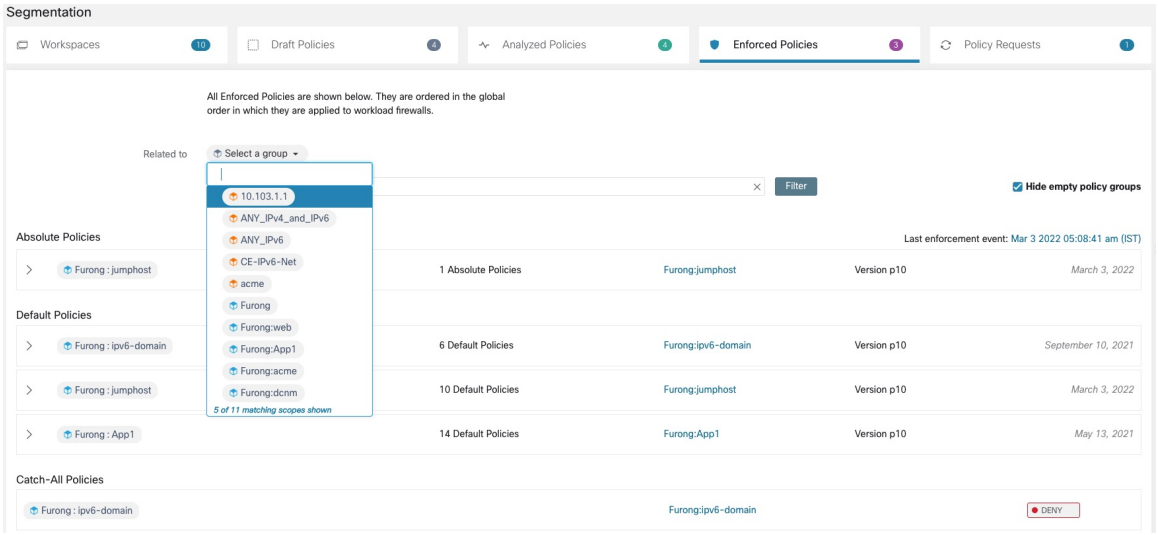
6. アプリのデフォルトポリシー
7. アプリ：コマースのキャッチオール

アプリ：人事の範囲とアプリ：コマースの範囲の相対的な順序は、2 つの範囲が重複する場合（つまり、両方の兄弟範囲に属するワークロードがある場合）にのみ問題になります。これは、ポリシーが常に範囲の下で定義されるためです。1 つの範囲のみに属するワークロードは、他の範囲のポリシーの影響を受けないため、順序は関係ありません。

ポリシーの順序と優先順位の検証

親/先祖ワークスペース内のポリシーの順序と優先順位を検証するには、[防御（Defend）]>[セグメンテーション（Segmentation）] ページの上部にある [分析されたポリシー（Analyzed Policies）] タブまたは [適用されたポリシー（Enforced Policies）] タブをクリックします。これらのビューは、それぞれ、分析されたポリシーと適用されたポリシーのグローバルビューを提供します。

図 *297*：例：ポリシーの優先度順の適用されたポリシーのリスト



- ポリシーのリストを、コンシューマまたはプロバイダーとして特定の範囲またはフィルタを含むポリシーのみに制限するには、範囲を選択するか、フィルタを入力します。
- 使用可能なフィルタ：

フィルタ名	定義
ポート（Port）	照合するポリシーのポート（例：80）。
[Protocol]	照合するポリシーのプロトコル（TCP など）。
承認済み（Approved）	[承認済み（Approved）] としてマークされているポリシーを照合します。承認済みポリシー （566 ページ）

フィルタ名	定義
外部？（External?）	コンシューマとプロバイダーが異なる範囲内にあるポリシー。
アクション（Action）	ポリシーアクション：許可または拒否（Allow または Deny）

（上級）ポリシーの優先順位の変更



注意 範囲ポリシーの優先順位の変更はほぼ必要ありません。ポリシーの優先順位を変更すると、すべてのワークスペースでの適用結果に影響を与える可能性があるため、変更は慎重に行ってください。

この機能へのアクセスは、サイト管理者などの非常に高い権限ロールを持つユーザーに制限されています。

始める前に

範囲の優先順位を変更する前：

- ポリシーの並べ替えロジックと、範囲におけるポリシーの優先順位が個々のポリシーインテントの順序付けにどのように変換されるかを理解します。ポリシーの優先順位（608ページ）を参照してください。
- 新しい順序が期待どおりだと確信できるまで、セカンダリワークスペースで変更を行います。
- 次のガイドラインを考慮して、変更を計画します。

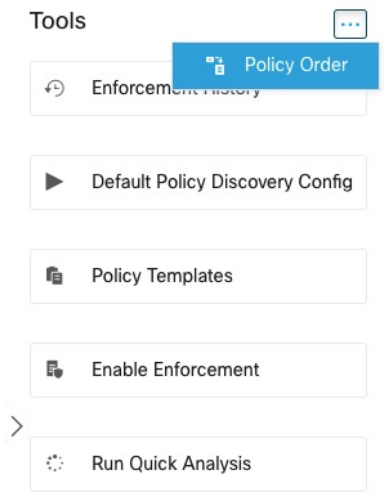
  並べ替えるときは、範囲ツリーの階層構造を利用するために、親優先の順序（親範囲が子範囲の上）を維持します。

  （兄弟範囲が重複している場合、兄弟範囲とその子の順序を変更する必要がある場合があります。兄弟範囲の重複は推奨されていないため、範囲クエリを更新して、重複を修正します。範囲の重複（430ページ）を参照してください。）

手順

ステップ 1 ポリシーの優先順位を並べ替えるには、[ツール（Tools）] の横にあるメニューアイコンをクリックし、[ポリシーの順序（Policy Order）] を選択します。

図 *298 : [*ポリシーの優先順位（***Policy Priorities***）*]* ページへの移動

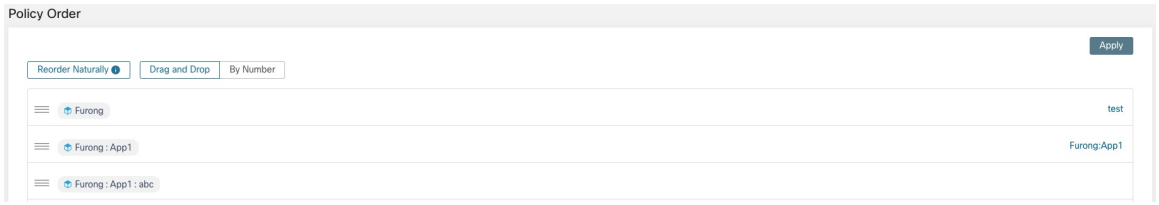


[ポリシーの順序（Policy Order）]ページでは、現在のポリシーの優先順位に従って、すべての範囲と範囲に対応するプライマリワークスペースのリストを確認できます。

ステップ 2 範囲を並べ替える方法は複数あります。

- リスト全体を並べ替えて、親範囲を子範囲の上に配置（「事前整列」）するには、[自然に並べ替える（Reorder Naturally）] をクリックします。これは推奨される順序であり、逸脱する場合は注意が必要です。
- リストを手動で並べ替えるには、次の手順を実行します。
  - 行を上下にドラッグします。
  - [番号順（By Number）] をクリックして、並べ替えに使用する各範囲の番号を設定します。この方法は、大きなリストの場合は簡単です。

図 *299 :* 範囲のポリシー優先順位の設定



次のタスク

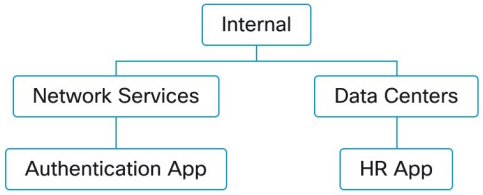
簡易分析を実行して、変更の結果を確認します。

コンシューマとプロバイダーが異なる範囲にある場合：ポリシーオプション

シナリオ例

次の状況は、クロス範囲トラフィックを示す例です。

範囲階層には、認証アプリケーション（プロバイダー）を含むネットワークサービス範囲が含まれています。範囲階層の別のブランチ上の範囲のメンバーである HR アプリケーションは、認証アプリケーションによって提供されるサービスのコンシューマです。



ポリシーオプション

Cisco Secure Workload は、この状況に対処するいくつかの方法を提供します。

オプション	手順	長所と短所
コンシューマとプロバイダーの両方を子または子孫として含む親または祖先の範囲でこれらのポリシーを作成します。	• 共通祖先範囲に 1 つ以上のポリシーを手動で作成します。 （オプション）より正確なポリシーを得るには、インベントリフィルタを使用してワークロードをグループ化します。例と手順については、「インベントリフィルタの作成（453ページ）」を参照してください。 • 範囲ツリーのブランチ全体に対して、共通の先祖範囲内のポリシーを自動的に検出します。	これらが、クロス範囲ポリシーに対処する最も簡単な方法です。 これらの方法では、コンシューマとプロバイダーのペアごとに 1 つのポリシーのみが必要です。 自動ポリシー検出の使用を検討している場合は、1 つの範囲または範囲ツリーのブランチのポリシーの検出（538ページ）の重要なその他の考慮事項を参照してください。

オプション	手順	長所と短所
クロス範囲ポリシーを作成するための高度な方法を使用する	個々の各範囲のポリシーを自動的に検出します。 （上級）クロス範囲ポリシーの作成（616ページ）を参照してください。 （この手順は、手動で作成されたポリシーと検出されたポリシーの両方に適用されます）	この方法では、コンシューマとプロバイダーのペアごとに、コンシューマ用のポリシーとプロバイダー用のポリシーの 2 つのポリシーが必要です。 この方法では、コンシューマポリシーとプロバイダーポリシーが異なる人によって所有されている場合にポリシーを作成できます。 1 つの範囲または範囲ツリーのブランチのポリシーの検出（538 ページ）のその他の考慮事項を参照してください。

（上級）クロス範囲ポリシーの作成

この手順では、クロス範囲ポリシー（コンシューマとプロバイダーが異なる範囲にあるポリシー）を作成する詳細な方法について説明します。この手順は、手動で作成されたポリシーと自動的に検出されたポリシーの両方に当てはまります。

この方法では、会話の両端で会話の発生を許可する必要があるため、コンシューマとプロバイダーのペアごとに 2 つのポリシーが必要です。

- コンシューマの範囲内のポリシーでは、プロバイダーとの会話を許可する必要があります。

  および

- プロバイダーの範囲内のポリシーでは、コンシューマとの会話を許可する必要があります。

この手順には、クロス範囲ポリシーを作成するために各範囲の所有者が実行する必要がある手順が含まれています。自分のアクセス権限で両方の範囲を変更できる場合は、すべての手順を実行できます。

始める前に

- クロス範囲トラフィックを処理するためのより簡単なオプションを検討してください。コンシューマとプロバイダーが異なる範囲にある場合：ポリシーオプション（615 ページ）を参照してください。
- この方法を使用するポリシーは、コンシューマとプロバイダーの両方のプライマリワークスペースで作成する必要があります。

  ポリシーで指定するプロバイダー範囲にプライマリワークスペースがない場合は、この方法を使用してクロス範囲ポリシーを作成する前にプライマリワークスペースを作成します。

- ポリシー要求を作成するには、ポリシーにALLOWアクションを含める必要があります。
- これらの要件に関する追加の詳細については、ポリシー要求（618ページ）を参照してください。
- （オプション）クロス範囲ポリシー要求の自動処理に関するオプションを検討します。クロス範囲ポリシー要求の自動処理（623ページ）を参照してください。
- （オプション）クロス範囲ポリシーを、範囲全体ではなく、コンシューマ範囲またはプロバイダー範囲内にあるクラスタ内のワークロードにのみ適用する場合は、クラスタをインベントリフィルタに変換する（601ページ）を参照してください。この手順を使用して作成されたクロス範囲ポリシーでは、クラスタを使用できません。

  ポリシーを自動的に検出している場合は、外部依存関係（549ページ）およびワークスペースの外部依存関係の微調整（551ページ）も参照してください。

手順

ステップ1 コンシューマのプライマリワークスペースで、手動で、または自動ポリシー検出を使用して、目的のポリシーを作成します。

作成したクロス範囲ポリシーごとに、プロバイダーに対するポリシー要求が自動的に作成されます。

ポリシー要求を表示するには、ポリシー要求の表示、承認、および拒否（618ページ）を参照してください。

注：プロバイダーアプリケーションのワークスペース内の既存のポリシーがこのトラフィックに一致する場合、新しいポリシーは不要なため、要求は作成されません。この状況は、解決済みのポリシー要求（627ページ）で説明されているように示されます。

ステップ2 ユーザー（またはプロバイダーアプリケーションの所有者）は、各ポリシー要求に応答する必要があります。

ポリシー要求の表示、承認、および拒否（618ページ）を参照してください。

ポリシー要求を受け入れると、プロバイダーのプライマリワークスペースに必要なポリシーが自動的に作成され、2つのアプリケーション間のトラフィックが許可されます。

要求元のアプリケーションからのトラフィックを許可しない場合、要求は拒否されます。

ステップ3 （オプション）ポリシーを自動的に検出している場合、ワークスペースの外部依存関係の微調整（551ページ）が必要な場合があります

ステップ4 両方のプライマリワークスペースを確認して分析します。

次のタスク

ポリシーを適用する準備ができたら、両方のプライマリワークスペースを適用する必要があります。

ポリシー要求

ポリシー要求は、「（上級）クロス範囲ポリシーの作成（616ページ）」で説明している方法を使用してクロス範囲ポリシーを作成すると生成されます。プロバイダーが別の範囲のメンバーであるときに、コンシューマ範囲のプライマリワークスペースにポリシーが作成されるたびに、プロバイダーの範囲に関連付けられたプライマリワークスペースにポリシーがまだ存在しない場合、ポリシー要求が生成されます。

このポリシー要求は、依存するアプリケーションに必要なサービスへのアクセスを許可するようプロバイダーアプリケーションの所有者に警告します。

「ポリシー要求の表示、承認、および拒否（618ページ）」および「クロス範囲ポリシー要求の自動処理（623ページ）」で、ポリシー要求を表示および応答するためのオプションを参照してください。

ポリシーリクエストに関するその他の詳細

- 提供されるサービスページ（ポリシー要求が表示される）は、プライマリワークスペースでのみ使用できます。これは、セカンダリワークスペースでの隔離された実験によって、他のプライマリワークスペースで通知が作成されないようにするためです。
- 外部範囲（ポリシーで指定されたプロバイダーがコンシューマとは異なる範囲に属している場合）にプライマリワークスペースがない場合、要求は送信されません（たとえば、ルート範囲の場合や、組織外のワークロードに対して定義された範囲の場合に該当する可能性があります）。外部範囲がポリシーを公開していない場合、ポリシーの分析と適用はコンシューマ側でのみ実行されます。
- プロバイダーがコンシューマとは異なる範囲にある場合、クラスタはサポートされません。ポリシーのコンシューマがクラスタの場合、ポリシー要求がコンシューマアプリケーションの範囲からのものであるかのように作成されます。プロバイダーからの同じサービスを消費する複数のポリシーをグループ化できます。
- ポリシー要求はプロバイダーに対してのみ生成され、コンシューマに対しては生成されません。コンシューマワークスペースがポリシーを分析または適用している場合、自動ポリシー検出または明示的な手動ポリシー作成によって、すべての正当な消費フローを許可するポリシーを明示的に含める必要があります（外部プロバイダーワークスペースからのポリシー要求は生成されません）。

ポリシー要求の表示、承認、および拒否

（上級）クロス範囲ポリシーの作成（616ページ）で説明されている方法を使用してクロス範囲ポリシーを作成する場合、コンシューマの範囲のポリシーに加えて、プロバイダーの範囲のプライマリワークスペースにもポリシーが必要です。コンシューマの範囲のプライマリワークスペースでクロス範囲ポリシーが作成されると、プロバイダーの範囲のプライマリワークスペースでポリシー要求が自動的に作成されます。

このトピックの情報を使用して、要求を受け入れる（プロバイダー範囲で必要なポリシーを作成する）か、要求を拒否します（拒否するとクロス範囲ポリシーは有効になりません）。

ポリシー要求を表示、承認、または拒否する方法：

目的	操作手順
すべてのポリシー要求を表示する	**1.** [防御（Defend）] > [セグメンテーション（Segmentation）] の順に選択します。 **2.** ポリシーページの最上部にある [ポリシー要求（Policy Requests）] をクリックします。 **3.** 特定のコンシューマ範囲をクリックして、その範囲からのポリシー要求を表示します。
特定の範囲のポリシー要求を表示する	プロバイダー範囲の保留中のポリシー要求を表示するには、次の手順を実行します。 **1.** [防御（Defend）] > [セグメンテーション（Segmentation）] の順に選択します。 **2.** 該当する範囲のプライマリワークスペースをクリックします。 **3.** [ポリシーの管理（Manage Policies）] をクリックします。 **4.** [提供されるサービス（Provided Services）] をクリックします。 タブに数値が表示されない場合、このワークスペースに保留中のポリシー要求はありません。 **5.** [ポリシー要求（Policy Requests）] をクリックします。 **6.** 特定のコンシューマ範囲をクリックして、その範囲からのポリシー要求を表示します。 または コンシューマ範囲からのポリシー要求を表示するには、次の手順を実行します。 コンシューマ範囲のプライマリワークスペースの [ポリシー（Policies）] タブで、[プロトコルとポート（Protocols and Ports）] 列の値をクリックし、ページの右側に表示されるパネルを確認します。[プロトコルとポート（Protocols and Ports）] セクションで、黄色のドットをクリックして、保留中のポリシー要求を表示します。

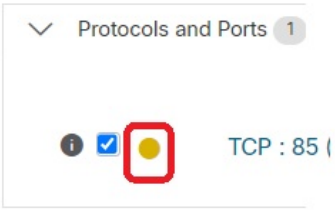
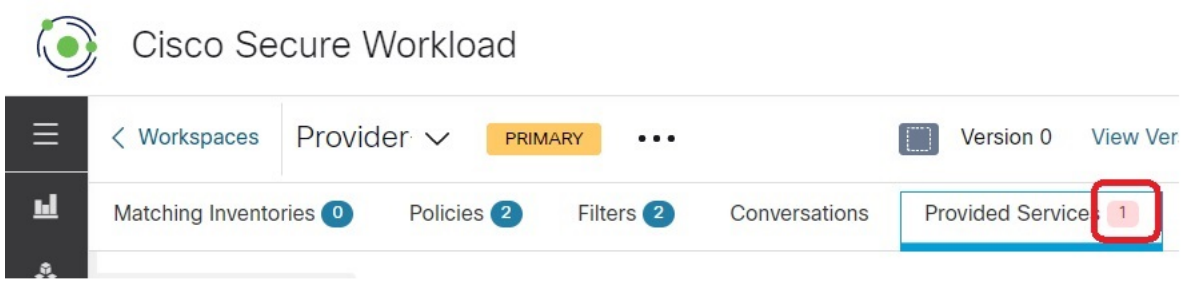
目的	操作手順
要求を手動で受け入れると、プロバイダー範囲に必要なポリシーが自動的に作成されます。	上記のいずれかの場所で、ポリシー要求の横にある [承認（Accept）] をクリックします。
要求を手動で拒否する	上記のいずれかの場所で、ポリシー要求の横にある [拒否（Reject）] をクリックします。
コンシューマワークスペースからポリシー要求ステータスを表示する	プライマリ コンシューマ ワークスペースの [ポリシー（Policies）] ページで、ポリシーをクリックしてから、ポート/プロトコルの値をクリックします。ステータスは、右側に開くパネルに表示されます。 保留中の要求は、黄色のドットで表示されます。  要求が承認されると、ドットが緑色のチェックマークに  変わります。 詳細についてはインジケータをクリックしてください。
プロバイダーのワークスペースからポリシー要求ステータスを表示する	上記の [提供されるサービス（Provided Services）] タブで要求ステータスを表示します。
ポリシー検出でプロバイダーに必要なポリシーを作成できるようにする	対応するフローが確実に表示される時間範囲を使用して、プロバイダー範囲のプライマリワークスペースでポリシーを自動的に検出し、ポリシーを公開します。
ポリシー要求の処理を自動化するためのオプションも参照してください	クロス範囲ポリシー要求の自動処理 （623 ページ）

図 *300* : プロバイダーのワークスペースで保留中のポリシー要求



ポリシー要求の承認：詳細

サービスでポリシー要求を承認することは、コンシューマとしての要求されたフィルタから、プロバイダーとしてのサービスに向けてポリシーを作成することと同じです。さらに、ポリシー要求を承認すると、コンシューマ アプリケーションのワークスペース（この例では、FrontEnd アプリと Serving Layer）からの元のポリシーが承認済みとしてマークされます（下の図を参照）。

図 *301* : ポリシー要求の承認/拒否

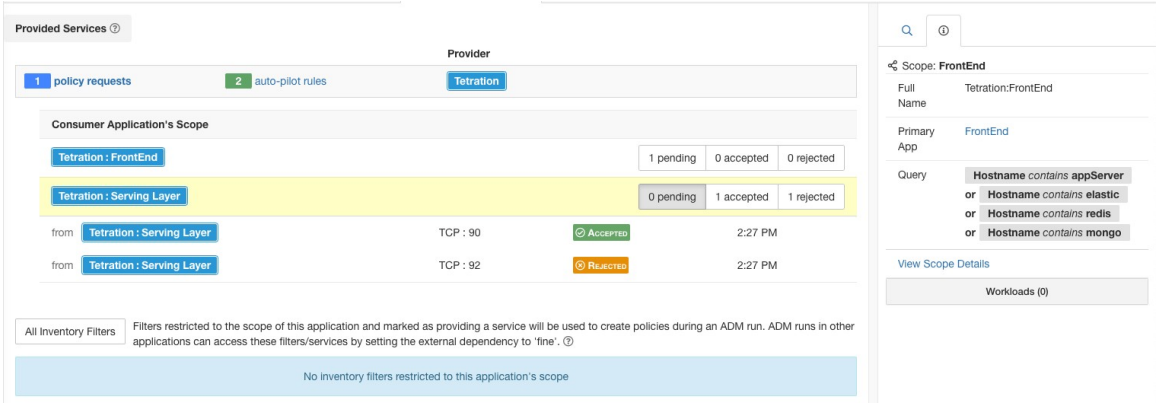
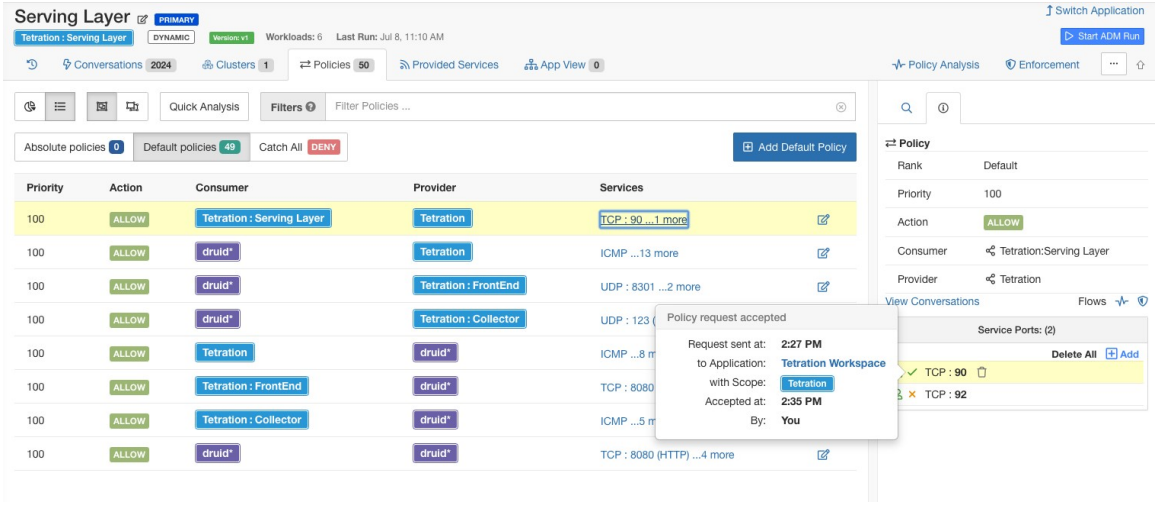


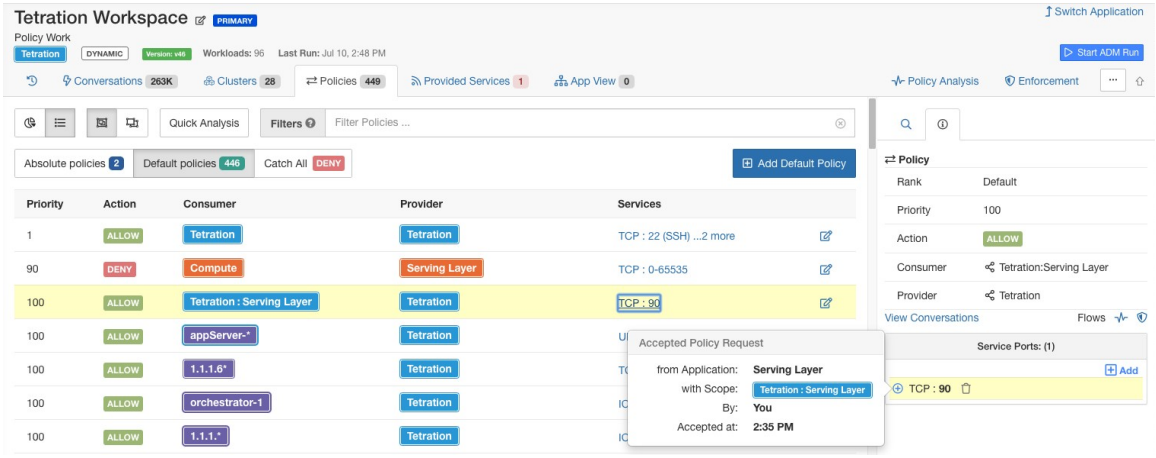
図 *302* : 承認済みとして表示されるポリシーのステータス



プロバイダー アプリケーションのワークスペース（この例では、ワークスペースの名前は Tetration）で作成された新しいポリシーには、このポリシーが外部ポリシー要求によって作成されたことを示す [プラス（plus）] アイコンが付いています。

（注） ポリシー要求が承認された後に、コンシューマ側の元のポリシーが削除された場合、プロバイダー側のポリシーは削除されません。ただし、ポリシーの横にあるツールチップには、元のポリシーがイベントのタイムスタンプとともに削除されたものとして表示されます。

図 *303* : ポリシー要求を承認することによって作成されたプロバイダー側のポリシー

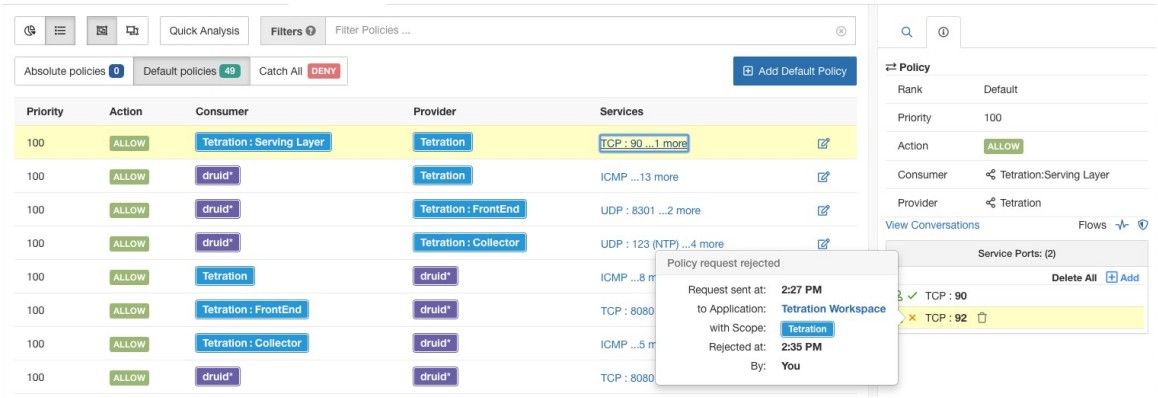


ポリシー要求の拒否：詳細

ポリシー要求を拒否した場合、いかなるポリシーも作成または更新されません。コンシューマ アプリケーションのワークスペース（この例では、Serving Layer アプリ）の元のポリシーは拒否されたことが示されますが、そのポリシーは有効なままです。つまり、アウトバウンドトラ

フィックは引き続き許可されます。拒否されたポリシーの横にあるツールチップには、プロバイダーアプリケーション、ポリシー要求を拒否したユーザー、拒否の時刻に関する情報が表示されます。

図 *304*：拒否されたと表示されるポリシーのステータス



クロス範囲ポリシー要求の自動処理

ポリシー要求は、「（上級）クロス範囲ポリシーの作成（616ページ）」で説明している方法を使用してクロス範囲ポリシーを作成すると生成されます。

クロス範囲ポリシーを作成するときに生成されるポリシー要求の数を減らすには、いくつかのオプションがあります。

表 *29*：ポリシー要求を自動的に処理するためのオプション

目的	操作手順
特定のコンシューマとプロバイダーのペア間のポリシー要求の処理を指定する	オートパイロットルール（624ページ）を参照してください。 必要な権限を持っている必要があります。
特定のワークスペースでのポリシー検出中に作成されたすべてのクロス範囲ポリシーについて、プロバイダーに必要なすべてのポリシーを自動的に作成する	自動ポリシー検出の実行を開始するときは、[詳細設定（Advanced Configurations）]セクションの[発信ポリシーコネクタを自動的に受け入れる（Auto accept outgoing policy connectors）]オプションを有効にします。 このオプションは、ルート範囲の所有者とサイト管理者が使用できます。 詳細については、次を参照してください。 自動ポリシー検出の詳細設定（554ページ）および 自動承諾ポリシーコネクタ（626ページ）

目的	操作手順
すべてのワークスペースからのすべてのポリシー要求のデフォルトの処理を指定する	[デフォルトのポリシー検出設定（Default Policy Discovery Config）] ページで、[詳細設定（Advanced Configurations）] セクションの [発信ポリシーコネクタを自動的に受け入れる（Auto accept outgoing policy connectors）] オプションを有効にします。 このオプションは、ルート範囲の所有者とサイト管理者が使用できます。 詳細については、次を参照してください。 デフォルトのポリシー検出設定 （564 ページ）および 自動ポリシー検出の詳細設定 （554 ページ）および 自動承諾ポリシーコネクタ （626 ページ）

オートパイロットルール

この機能は、「（上級）クロス範囲ポリシーの作成（616ページ）」で説明されている方法を使用してクロス範囲ポリシーを作成する場合にのみ適用できます。

データセンター内の他の多くのアプリケーションにサービスを提供するインフラストラクチャアプリケーションは、他のアプリケーションから多数のポリシー要求を受信する可能性があります。

将来の一致するポリシー要求を自動的に受け入れるか拒否するオートパイロットルールを作成することで、ポリシー要求の数を減らすことができます。

（注） オートパイロットルールは、既存のポリシー要求には適用されません。将来のポリシー要求にのみ影響します。

オートパイロットルールを使用してポリシー要求を自動的に受諾または拒否する

指定されたポートで、指定されたコンシューマとプロバイダーのペア間のポリシー要求を自動的に受け入れるか拒否するようにオートパイロットルールを構成します。オートパイロットルールは、広範囲（範囲間）にすることも、各範囲内のワークロードのサブセットにのみ適用することもできます。サブセットは、インベントリフィルタによって構成されます。コンシューマ、プロバイダー、またはそれぞれに対してインベントリ フィルタを使用できます。

1. オートパイロットルールを範囲全体ではなく、範囲内のワークロードのサブセットに適用する場合は、次のようにします。

   関連する範囲でインベントリフィルタを作成して、ワークロードをグループ化します。範囲のメンバーであるワークロードのみがフィルタに含まれるように、各インベントリフィルタで [クエリを所有権の範囲に制限する（Restrict Query to Ownership Scope）] オプションが選択されていることを確認してください。

2. [防御（Defend）] > [セグメンテーション（Segmentation）] を選択します。
3. 特定のプロバイダーに関連するポリシー要求を自動的に受け入れるか拒否するコンシューマ範囲のプライマリワークスペースをクリックします。
4. [ポリシーの管理（Manage Policies）] をクリックします。
5. [提供されるサービス（Provided Services）] をクリックします。
6. インベントリフィルタに対してこのルールを作成する場合は、目的のインベントリフィルタに対して次の手順を実行します（インベントリフィルタはオレンジ色のアイコンで識別されます）。

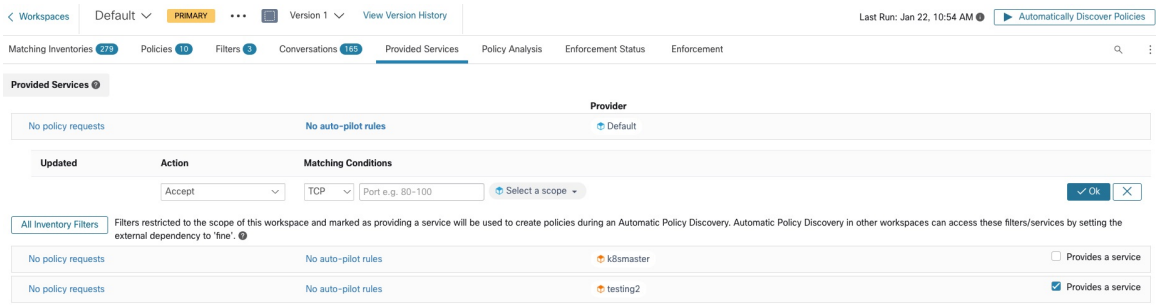
   それ以外の場合は、範囲に対してこれらの手順を実行します（範囲は青いアイコンで識別されます）。

   正しい場所をクリックしていることを確認してください。
7. [オートパイロットルールなし（No Auto-Pilot Rules）] または [オートパイロットルール（Auto-Pilot Rules）] のいずれか表示されている方をクリックします。
8. [新規オートパイロットルール（New Auto-Pilot Rule）] をクリックします。
9. オートパイロットルールを構成します。プロバイダーを表す範囲またはインベントリフィルタを選択します。
10. [OK] をクリックします。

オートパイロットルールの例

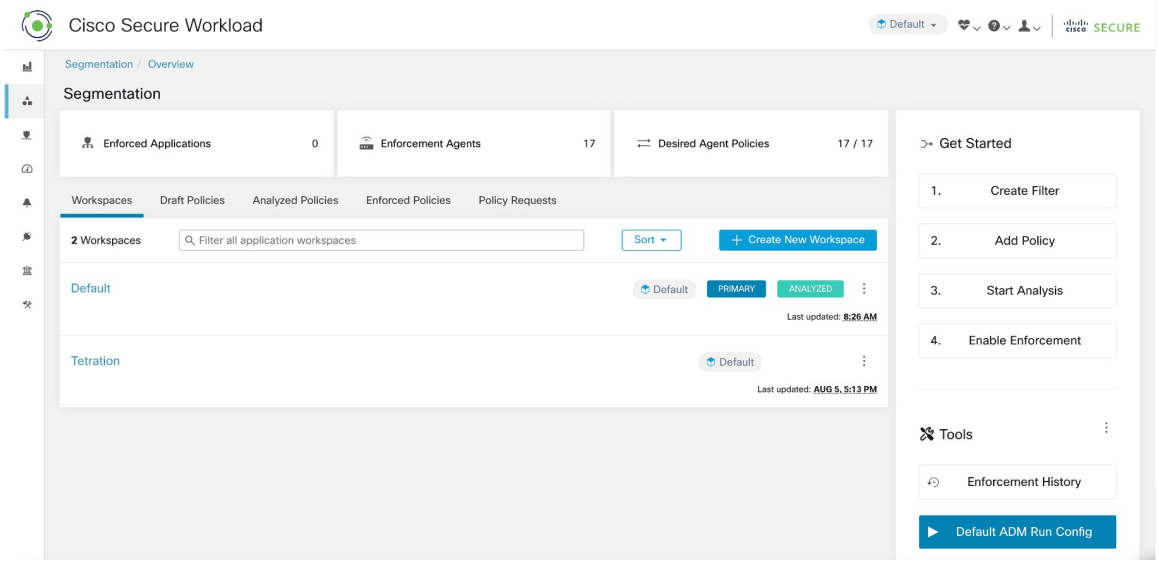
以下の例では、Tetration:Adhoc に含まれる任意のコンシューマからプロバイダーサービス Tetration への、ポート範囲 1 ～ 200 における TCP ポリシー要求を拒否する新しいオートパイロットルールを作成します。

図 *305* : オートパイロットルールの作成/更新



次に、FrontEnd アプリケーションのワークスペースに TCP ポート 23 における新しいポリシーを作成します。ポリシーはオートパイロットルールに一致するため、自動的に拒否されます。ポリシー拒否のステータスと理由が、拒否されたポリシーの横のツールチップに表示されます。

図 *306* : オートパイロットルールによってポリシーが自動的に拒否される



オートパイロットルールによって最近作成されたポリシーの数の表示

ワークスペースに対してライブポリシー分析が最後に開始（または再開始）されてから、オートパイロットルールによってワークスペース内に作成されたポリシーの数を表示するには、次の手順を実行します。

関連するプライマリワークスペースの [提供されるサービス（Provided Services）] ページに移動し、「自動作成された」ポリシーの数を探します。

自動承諾ポリシーコネクタ

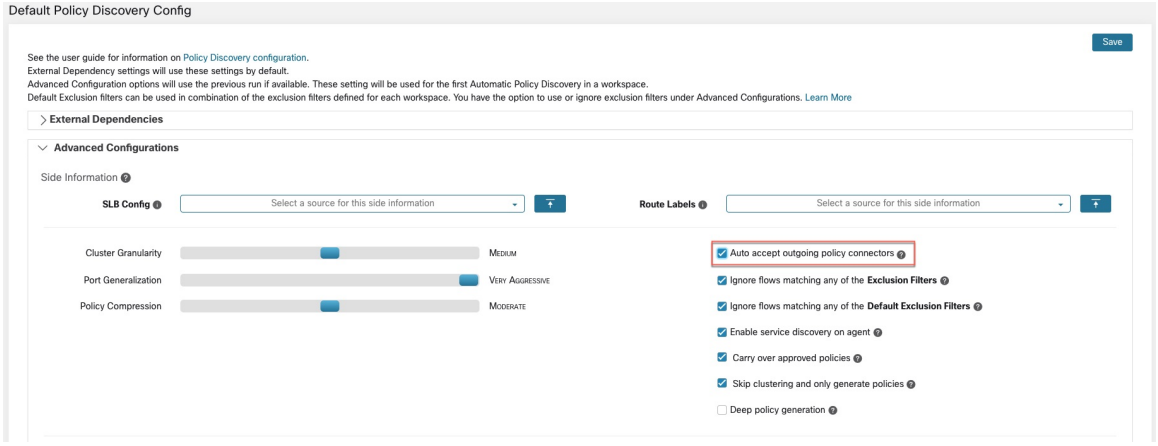
このオプションは、デフォルトのポリシー検出構成として設定するか、各ワークスペースの自動ポリシー検出の詳細オプションで設定できます。

自動ポリシー検出構成ページの [発信ポリシーコネクタを自動的に受け入れる（Auto accept outgoing policy connectors）] オプションを使用すると、自動ポリシー検出の一部として作成されたポリシー要求を自動的に受け入れることができます。

このオプションがデフォルトの自動ポリシー検出設定で有効になっている場合、手動で作成されたポリシー要求またはワークスペースのインポートによって作成されたポリシー要求も自動的に受け入れられます。

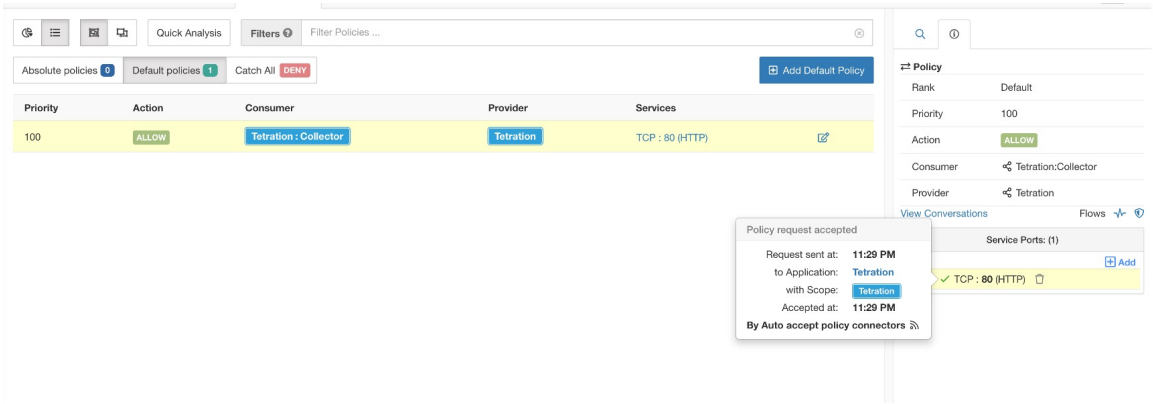
（注） このオプションは、ルート範囲の所有者またはサイト管理者のみが使用できます。

図 *307* : 発信ポリシーコネクタの自動受け入れオプション



このオプションを設定すると、ルート範囲の任意のワークスペースまたは該当するワークスペースで作成されたポリシー要求は、すべて自動的に受け入れられます。

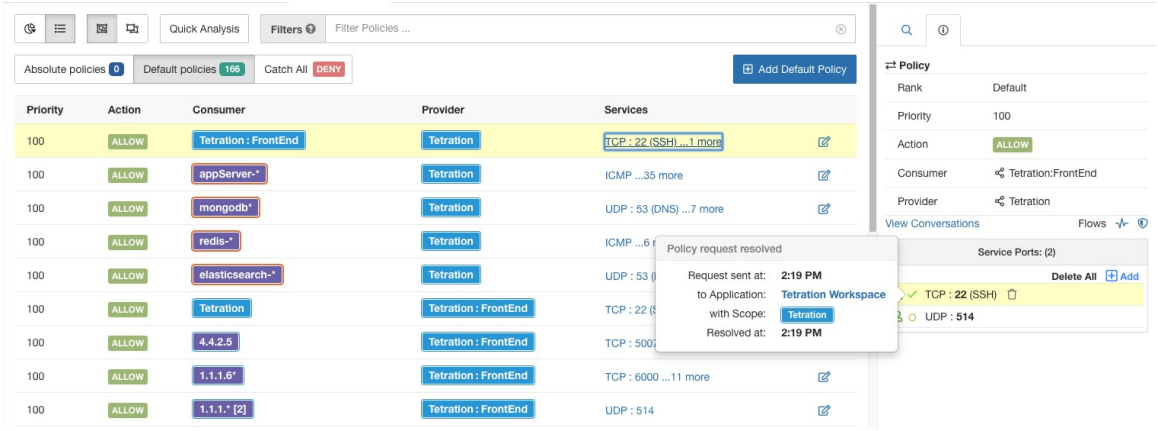
図 *308* : ポリシーはポリシーコネクタの自動受け入れによって自動的に受け入れられる



解決済みのポリシー要求

ポリシー要求を作成するためのすべての条件が満たされているが、プロバイダーアプリケーションのワークスペースに既存の一致するポリシーが存在する場合、コンシューマアプリケーションのワークスペースで作成されたポリシーは解決済みとしてマークされ、プロバイダーアプリケーションのワークスペースが要求されたポートを介してトラフィックをすでに許可していることを示します。

図 *309* : ポリシーのステータスが解決済みとして表示されている



提供されるサービス

このページは、コンシューマとプロバイダーが異なる範囲にあるポリシーを作成する場合、および（上級）クロス範囲ポリシーの作成（616ページ）で説明されている方法を使用している場合にのみ使用されます。

このページのオプションの詳細については、

- ポリシー要求（618ページ）
- オートパイロットルール（624ページ）
- インベントリフィルタの作成（453ページ）および外部依存関係（549ページ）（[サービスの提供（Provides a service）] オプションに関する情報）を参照してください。

このページにアクセスするには、プライマリワークスペースに移動し、[ポリシーの管理（Manage Policies）]、[提供されるサービス（Provided Services）] の順にクリックします。

クロス範囲ポリシーのトラブルシューティング

（上級）クロス範囲ポリシーの作成（616ページ）で説明されている方法を使用してクロス範囲ポリシーを作成した場合は、コンシューマワークロードとプロバイダーワークロードのそれぞれのプライマリワークスペースに、トラフィックを許可するポリシーが必要です。必要なポリシーが両方のワークスペースに存在することを確認します。

いずれかのポリシーが削除または変更されても、通知は行われません。

ポリシーの検出中にポリシーペアが生成された場合は、ポリシーの承認に関する情報を参照して、後続の検出の実行から保護します。ポリシーの承認（566ページ）を参照してください。

（上級）クロス範囲ポリシーの作成（616ページ）に記載されている他の要件が引き続き満たされていることを確認します。

必要なポリシーを持つコンシューマとプロバイダーの両方のワークスペースを適用する必要があります。

クロス範囲ポリシーに役立つツール

- ポリシーを検出した範囲とは異なる範囲にプロバイダーが存在するポリシーを見つけるには、[外部？（External?）] フィルタオプションを使用します。
- ポリシーのビジュアルビューには、外部ポリシーを表示するオプションがあります。ポリシーの視覚的表現 （637 ページ）を参照してください。

デフォルトのポリシー検出設定を使用している場合

デフォルトの外部依存関係の設定を個々のワークスペースで使用できるように変更した後、[デフォルトのポリシー検出設定（Default Policy Discovery Config）] ページで [保存（Save）] をクリックしたことを確認します。

有効なコンシューマまたは有効なプロバイダー

ポリシーで指定されたコンシューマとプロバイダーによって、次のことが決まります。

- ポリシーを受信する Secure Workload エージェントを含む一連のワークロード。
- インストールされているファイアウォールルールの影響を受ける一連の IP アドレス。

デフォルトでは、これらは同じです。

ただし、ポリシーを受信するワークロードの IP アドレスとは異なる IP アドレスのグループを、ファイアウォールルールで指定する必要が生じることがあります（以下の例を参照してください）。

このニーズに対処するために、有効なコンシューマや有効なプロバイダーを設定できます。

コンシューマとプロバイダーのデフォルトの動作

デフォルトでは、Secure Workload エージェントがポリシーを受け取ると、ファイアウォールルールはそのワークロードに固有のものになります。次の例でこれを詳しく説明します。

1.1.1.0/24 サブネットが指定されたプロバイダーフィルタを持つ許可ポリシーがあるとします。このポリシーが IP アドレス 1.1.1.2 のワークロードでプログラムされている場合、ファイアウォールルールは次のようになります。

- 着信トラフィックの場合、ファイアウォールルールは、サブネット 1.1.1.0/24 全体ではなく、厳密に 1.1.1.2 宛てのトラフィックのみを許可します。
- 発信トラフィックの場合、ファイアウォールルールは、サブネット 1.1.1.0/24 全体からではなく、厳密に 1.1.1.2 からのトラフィックのみを許可します（スプーフィングを防ぐため）。

必然的に、1.1.1.0/24 サブネット内に IP アドレスを持たないワークスペースに属するエージェントのワークロードは、上記のファイアウォールルールを受け取りません。

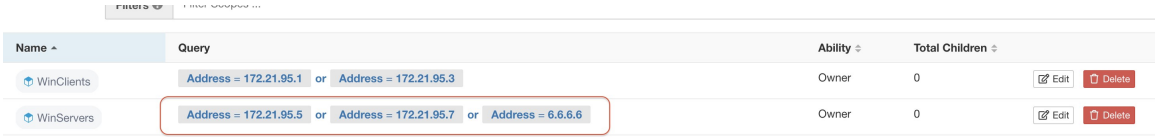
例：有効なコンシューマまたは有効なプロバイダー

この例では、仮想 IP（VIP）の背後にある一連のワークロードのポリシーを設定します。これは、キープアライブや Windows フェールオーバー クラスタリング ソリューションと似ています。有効なコンシューマや有効なプロバイダーを使用して、フェールオーバーイベント中にトラフィックが中断されないようにします。

IP アドレス（*172.21.95.5* および *172.21.95.7*）を持ち、VIP - *6.6.6.6* の背後でサービスを提供する一連のワークロードについて考えてみます。この VIP はフローティング VIP であり、任意の時点で 1 つのワークロードのみが VIP を所有します。目標は、フリート内のすべてのワークロードでファイアウォールルールをプログラムし、6.6.6.6 へのトラフィックを許可することです。

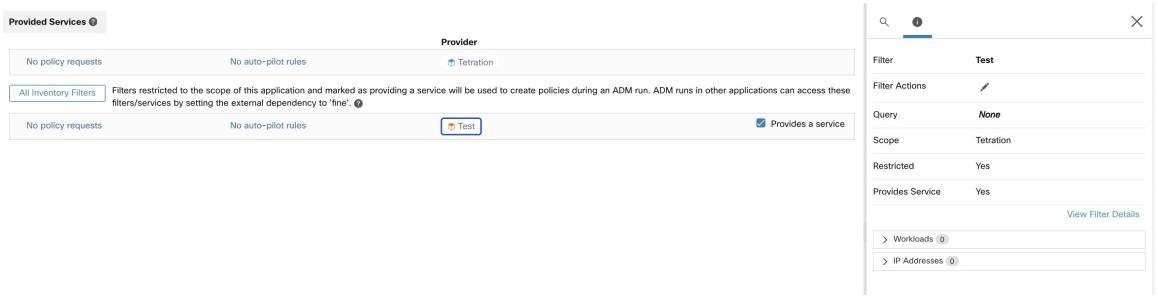
このセットアップでは、フリートを表すワークロードのクラスタ（172.21.95.5 および 172.21.95.7）と VIP（6.6.6.6）を含む範囲と対応するワークスペースがあります。

図 *310* : *VIP* とワークロードのクラスタで構成される範囲



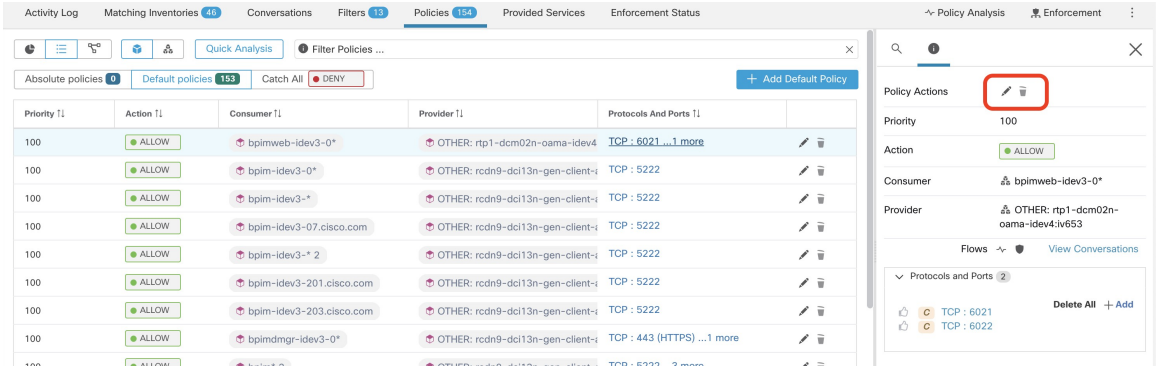
以下に示すように、VIP は提供サービスとしてこのワークスペースで公開されます。

図 *311* : 提供サービスとして公開された *VIP*



このサービスのクライアントからサービス VIP にポリシーを追加することにした場合、（デフォルトでは）VIP へのトラフィックを許可するファイアウォールルールは、VIP を所有するワークロードでのみプログラムされます。ただし、フェールオーバーイベントが発生すると、その後サービス VIP を所有する新しいワークロードが適切なファイアウォールルールを取得するまでに時間がかかり、トラフィックが短時間中断される場合があります。

図 *312 :* クライアントからサービス *VIP* へのトラフィックを許可するポリシー



この問題に対処するために、（以下の手順を使用して）有効なプロバイダーを設定します。具体的には、[有効なプロバイダー（Effective Provider）] の設定で、サービス VIP へのトラフィックを許可するファイアウォールルールのプログラミングが必要な一連のワークロードを指定します。これらのワークロードのいずれかが VIP を所有しているかどうかは関係ありません。

[有効なプロバイダー（Effective Provider）] が設定されている場合、ワークロードが VIP を所有していない場合でも、6.6.6.6 へのトラフィックを許可するファイアウォールルールがワークロード上でプログラムされていることを確認できます。サービスをサポートするすべてのワークロードをこれらのルールでプログラムする場合、新しいプライマリワークロード（VIP を所有）には必要なファイアウォールルールがプログラムされるため、フェールオーバーイベント中にトラフィックが中断されることはありません。

図 *313 : VIP* サービスへのトラフィックを許可するホストのファイアウォールルール

```
$
$ hostname -I | awk '{print $1}'          IP Address of
172.21.95.7                               the server
$                                         part of cluster
$
$ sudo iptables -n --list TA_INPUT   <— Ingress rules
Chain TA_INPUT (1 references)
target     prot opt source               destination
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           match-set ta_6c6b4133313438ff5429ca8c14b6 src match-set ta_ac2618d307e4e7dbb76b96c0df3f dst mul
tiport dports 1443 ctstate NEW,ESTABLISHED /* PolicyId=DEFAULT:100:ALLOW:5ed53fe8497d4f26444d50b3:5ed5435b497d4f26414d50b1:6 */
RETURN     all  --  0.0.0.0/0            0.0.0.0/0
$
$
$ sudo iptables -n --list TA_OUTPUT   <— Egress rules
Chain TA_OUTPUT (1 references)
target     prot opt source               destination
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           match-set ta_ac2618d307e4e7dbb76b96c0df3f src match-set ta_6c6b4133313438ff5429ca8c14b6 dst mul
tiport sports 1443 ctstate ESTABLISHED /* PolicyId=DEFAULT:100:ALLOW:5ed53fe8497d4f26444d50b3:5ed5435b497d4f26414d50b1:6 */
RETURN     all  --  0.0.0.0/0            0.0.0.0/0
$
$
$ sudo ipset list ta_ac2618d307e4e7dbb76b96c0df3f
Name: ta_ac2618d307e4e7dbb76b96c0df3f
Type: hash:net
Revision: 3
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 16816
References: 2
Members:
6.6.6.6              <— VIP
$ sudo ipset list ta_6c6b4133313438ff5429ca8c14b6
Name: ta_6c6b4133313438ff5429ca8c14b6
Type: hash:net
Revision: 3
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 16848
References: 2
Members:
172.21.95.1
172.21.95.3          <— Client IPs
$
```

有効なコンシューマまたは有効なプロバイダーの設定方法

1. 編集するポリシーをクリックします。
2. ポリシーの右上にある [編集（Edit）] ボタンをクリックして、詳細ポリシーオプションに移動します。
3. [有効なコンシューマ（Effective Consumer）] または [有効なプロバイダー（Effective Provider）] をクリックします。
4. 目的のアドレスを指定します。
5. 有効なコンシューマと有効なプロバイダーの両方のアドレスを指定する必要がある場合があります。

ポリシーの削除について

重要　ポリシーを削除する前に、コンシューマとプロバイダーが異なる範囲内にある場合、そのポリシーが必要なポリシーのペアの 1 つではないことを確認してください。

これを確認するには、[プロトコルとポート（Protocols and Ports）] 列でポリシーのリンクをクリックします。ページの右側に表示されるパネルで、[プロトコルとポート（Protocols and Ports）] セクションを確認します。クロス範囲ポリシー要求を受け入れることによって作成されたポリシーは、ポートとプロトコルの横のプラス記号で示されます。



プラス記号をクリックすると、クロス範囲ポリシーの作成者と、対応するコンシューマポリシーへのリンクが表示されます。

（注）　自動ポリシー検出によって提案され、承認されていないポリシーは、それらを生成したトラフィックフローが後続のポリシー検出の実行中に見られない場合、後続の実行後に存在しなくなる可能性があります。提案されたポリシーを保持するには、ポリシーの承認 （566 ページ）を参照してください。

ポリシーの確認と分析

ポリシーを適用する前に、ポリシーに意図した効果があること（また、意図していない効果がないこと）を確認することが不可欠です。

自動検出されたポリシーの確認

ポリシーを検出したワークスペースの [ポリシー（Policies）] ページでポリシー検出結果を確認します。

最初に確認すること

最初に、ポリシーで次の各領域が対処されているかどうかを、この推奨される順序で確認することを推奨します。

- 重要な一般ポート
- インターネットに接続するトラフィック
- 異なるアプリケーション間のトラフィック（これらのフローには、異なる範囲内のワークロードが関係する場合があります）
- 同じアプリケーション内のトラフィック（これらのフローには、同じ範囲内のワークロードが関係する可能性が高いです）

ポリシーの確認に役立つツール

- この作業をより管理しやすくするには、ポリシーをフィルタしてソートし、関連するポリシーをグループとして確認できるようにします。
  - コンシューマ、プロバイダー、ポート、プロトコルなどで列をソートするには、テーブルの見出しをクリックします。
  - 特定のサブセットを表示するには、ポリシーリストの上部にあるフィルタを使用します。

    フィルタに使用できるプロパティのリストを表示するには、[ポリシーのフィルタ（Filter Policies）] ボックスの [(i)] ボタンをクリックします。

- 生成されたポリシーのグラフィカル表示を確認します。

  [ポリシービジュアルビュー（Policy Visual View）] ボタン（  ）をクリックします。

  詳細については、「ポリシーの視覚的表現 （637 ページ）」を参照してください。

- ポートに基づいて行を検索またはフィルタするには、[グループ化解除（Ungrouped）] ボタンをクリックします。

- デフォルトでは、ポリシーは、コンシューマ、プロバイダー、アクションでグループ化されています。このビューに戻るには、[グループ化（Grouped）]ボタンをクリックします。
- ポリシーを検出した範囲とは異なる範囲にプロバイダーが存在するポリシーを見つけるには、[外部？（External?）] フィルタオプションを使用します。

  コンシューマとプロバイダーが異なる範囲にある場合：ポリシーオプション（615ページ）で説明されている方法のいずれかを使用して、このトラフィックのポリシーを作成します。
- 生成されたポリシーの信頼度レベルを確認します。信頼度の低いポリシーへの対処 （635ページ）を参照してください。
- ワークロードの詳細については、[ワークロードプロファイル（Workload Profile）] を参照してください。IP アドレスをクリックし、右側のペインで [ワークロードプロファイルの表示（View Workload Profile）] をクリックします。
- 特定のポリシーの生成に使用されたトラフィックフローを表示するには、そのポリシーの [プロトコルとポート（Protocols and Ports）] 列の値をクリックし、開いたサイドパネルで [カンバセーションの表示（View Conversations）] をクリックします。

  詳細については、カンバセーション （679 ページ）を参照してください。

  必要に応じて、[フロー検索（Flow Search）] をクリックしてさらにドリルダウンし、カンバセーションのフローを表示することができます。

その他の必要な作業と確認

- 不明な IP アドレス（フェールオーバーやその他のフローティング IP など）を識別し、それらが何であるかがわかるようにラベルでタグ付けします。

  [インベントリプロファイル（Inventory Profile）] ページで、役立つ詳細を確認できます。IP アドレスをクリックし、右側のペインで [インベントリプロファイルの表示（View Inventory Profile）] をクリックします。
- 明らかに望ましくない、または理にかなっていないものを探します。
- インベントリフィルタを使用してワークロードをグループ化し、単一のポリシーで複数のワークロードに対応できるようにします。インベントリフィルタの作成 （453ページ）を参照してください。
- 必要に応じて、他のネットワーク管理者を調べて連絡し、表示されるポリシーの必要性を把握します。
- ポリシーの複雑さの対処 （607ページ）の下のトピックも参照してください。各トピックには、手動および承認済みのポリシー、および自動的に検出されたポリシーが含まれます。
- 一般的に、範囲内のポリシーの最大数は、約500個以下にすることを推奨します。これよりも多くのポリシーがある場合は、同様のポリシーを統合できるかどうかを確認するか、範囲を分割することを検討してください。

- 確認しながら、正しいことがわかったポリシーをそのまま承認して、今後の検出の実行で保持します。

信頼度の低いポリシーへの対処

自動ポリシー検出の後、信頼度の評価によって、ポリシーで指定した各サービス（ポートとプロトコル）について、検出された各ポリシーの正確度と適切性が示されます。

検出された信頼度の低いポリシーを特定するには、次の手順を実行します。

1. 該当する範囲とワークスペースに移動し、[ポリシーの管理（Manage Policies）] をクリックします。
2. [Policies] タブをクリックします。
3. [グループ化されていないポリシーリストビュー（Ungrouped Policy List View）] ボタンをクリックします。
4. [信頼度（Confidence）] 列見出しをクリックして、ポリシーリストを信頼度レベル順に並べ替えます。
5. [プロトコルとポート（Protocols and Ports）] 列の値をクリックして、ウィンドウの右側にパネルを開きます。
6. [プロトコルとポート（Protocols and Ports）] セクションでは、指定した各サービス（ポートとプロトコル）の信頼度がそれぞれの [C] の色で示されます。

   信頼度の説明を表示するには、[C] にカーソルを合わせます。
7. リスト内でサービスの信頼度の低い指標を探します。
8. 該当する場合は、不要なポリシーを削除または編集するか、ポリシーを追加します。

特定のポリシーの信頼度レベルを表示するには、次の手順を実行します。

1. [ポリシー（Policies）] タブで、そのポリシーの [プロトコルとポート（Protocols and Ports）] 列の値をクリックします。

   ウィンドウの右側に [ポリシーサイドビュー（Policy Side View）] パネルが開きます。
2. [プロトコルとポート（Protocols and Ports）] セクションでは、指定した各サービス（ポートとプロトコル）の信頼度がそれぞれの [C] の色で示されます。

   信頼度の説明を表示するには、[C] にカーソルを合わせます。

フローの方向とポリシーの信頼度

検出されたポリシーの正確度は、フローの方向が正しく識別されたかに左右されます。フローの方向が正しく識別されていない場合、自動ポリシー検出結果の信頼度が低下する場合があります。ポリシー作成のために分析される通信フロー方向の決定については、「クライアントサーバーの分類」を参照してください。

自動ポリシー検出結果のトラブルシューティング

自動ポリシー検出結果が予期したものでない場合は、次の点を確認します。

選択した時間範囲を拡張してより多くのデータを含める

時間枠を拡張してより多くのデータを含め、まれに発生するイベントをキャプチャします。たとえば、アプリケーションが複数のプロバイダーアプリケーションから取得したデータを使用して複雑な四半期レポートを生成する場合は、そのトラフィックが含まれている時間範囲が含まれていることを確認します。

特定の変更の前にデータが収集されないようにする

範囲の定義が変更された場合、または特定の時間より前に収集されたデータが何らかの理由で無効になった場合は、時間範囲にその時点より前のデータが含まれていないことを確認します。

紛らわしいトラフィックフローを除外する

除外フィルタを設定または変更する必要が生じる場合があります。

除外フィルタは複数の場所で設定でき、複数の場所で有効または無効にできます。それぞれの場所を確認します。

- ワークスペースに設定されている除外フィルタを確認します。
- [デフォルトのポリシー検出設定（Default Policy Discovery Config）] ページの下部で設定されている、デフォルトの除外フィルタを確認します。
- 自動ポリシー検出のワークスペースの設定の [詳細設定（Advanced Configurations）] セクションで、どの除外フィルタが有効になっているかを確認します。
- [デフォルトのポリシー検出設定（Default Policy Discovery Config）] ページの [詳細設定（Advanced Configurations）] セクションで、どの除外フィルタが有効になっているかを確認します。
- デフォルトの除外フィルタを使用する場合は、[デフォルトのポリシー検出設定（Default Policy Discovery Config）] ページで [保存（Save）] をクリックして、これらの設定を個々のワークスペースで使用できるようにしていることを確認します。

詳細については、除外フィルタ （545 ページ）およびサブトピックを参照してください。

コンシューマとプロバイダーが異なる範囲内にあるポリシーのトラブルシューティング

クロス範囲ポリシーのトラブルシューティング （628 ページ）を参照してください。

承認済みポリシーのステータスの確認

承認されたポリシーのトラブルシューティング （567 ページ） を参照してください。

ポリシーの視覚的表現

ポリシーの視覚的表現により、ポリシーのグラフィカルビューが可能になります。

ポリシーの視覚的表現のページに移動するには、[ポリシー（Policies）]ページで、リストアイコンの右側にあるグラフアイコン（  ）をクリックします。

ポリシービューの要素

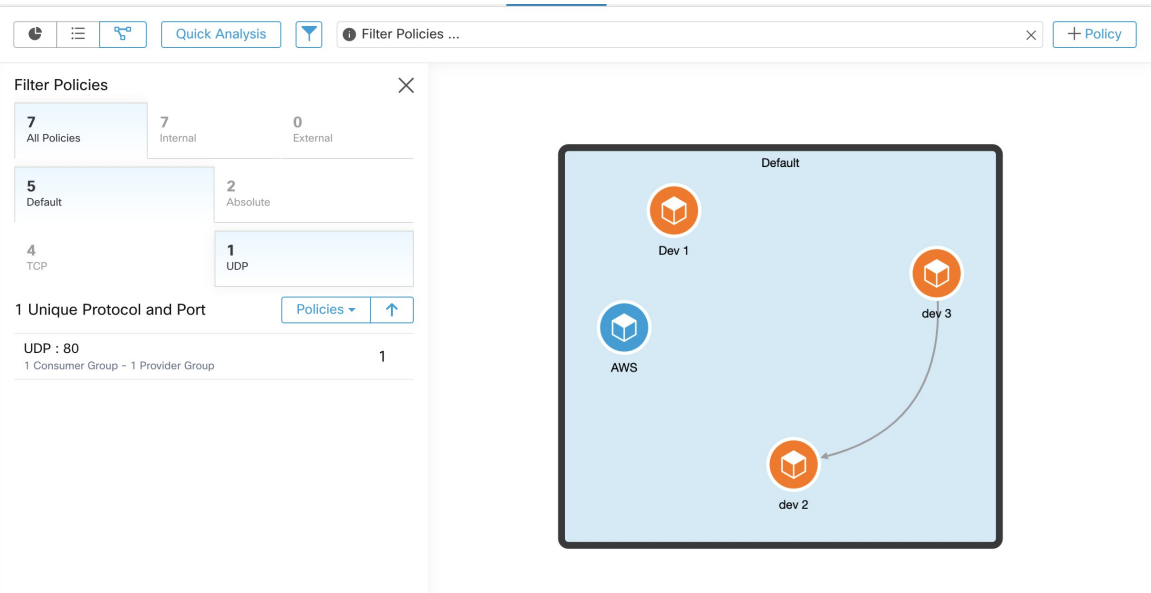
ポリシービューの視覚的な要素は次のとおりです。

要素	表す内容
青、オレンジ、または紫のアイコン	ノード（ポリシーのコンシューマまたはプロバイダー）
青のアイコン	範囲
オレンジのアイコン	インベントリフィルタ
紫のアイコン	クラスタ
2 つのアイコンをつないでいる線	1 つ以上のポリシー

ポリシービューのオプション

目的	操作手順
コンシューマノードまたはプロバイダーノードに含まれるワークロードのリストを表示する	ノードのアイコンをダブルクリックします。
サービス（ポート）、アクション（許可/拒否）、およびコンシューマとプロバイダー間のプロトコルなどのポリシーの詳細を表示する	それらをつないでいる線をダブルクリックします。右側のペインに詳細が表示されます。
ノードに送受信されるポリシーを表示する	アイコンをクリックします。
範囲内のワークロード間のポリシーのみを表示する	[内部（Internal）] ボタンをクリックします。
プロバイダーがコンシューマとは異なる範囲内にあるポリシーのみを表示する	[外部（External）] ボタンをクリックします。
詳細なフィルタリングオプションを使用する	フィルタテキスト入力ボックスの左側にある [(i)] ボタンをクリックしてオプションを表示し、フィルタ条件を入力します。

図 *314* : グラフィカルビューでのポリシーのフィルタリング



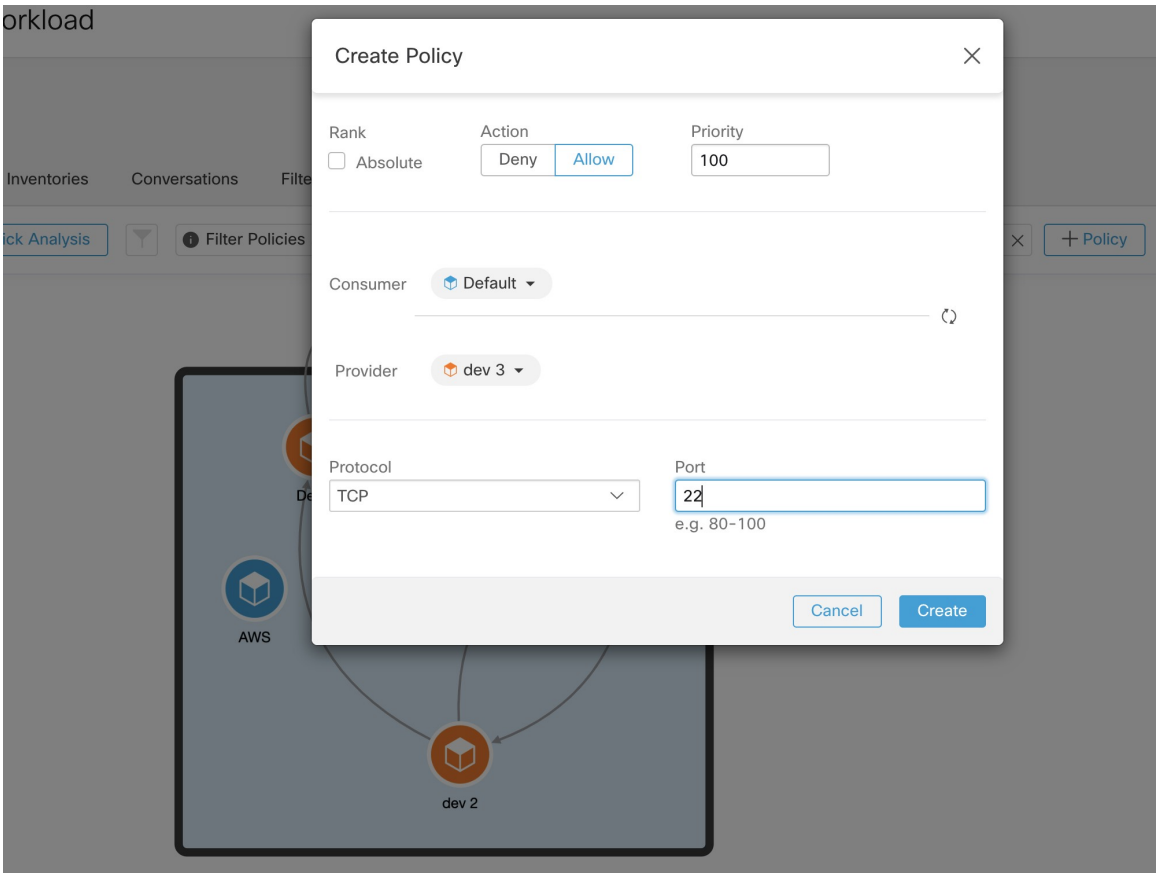
ポリシーのグラフィカルビューの高解像度画像をダウンロードするには、次の手順を実行します。

1. グラフの右下隅にある省略記号アイコンをクリックし、[画像のエクスポート（Export Image）] をクリックします。
2. 必要な解像度と画像のタイプを選択します。
3. [ダウンロード（Download）] をクリックします。

ポリシーの追加（ポリシービューページ）

ポリシーを作成するには、コンシューマにカーソルを合わせ、「+」記号が表示されたら、ポリシーをクリックしたままプロバイダーにドラッグします。絶対ポリシーを作成するには、モーダルの [絶対（Absolute）] チェックボックスをオンにします。それ以外の場合、ポリシーはデフォルトポリシーとして作成されます。ポリシーは、線をクリックして、ポップアップリストからポリシーを選択することによっても管理できます。ポリシーがサイドバーに表示されます。

図 *315*: グラフィカルビューでのポリシー作成



簡易分析

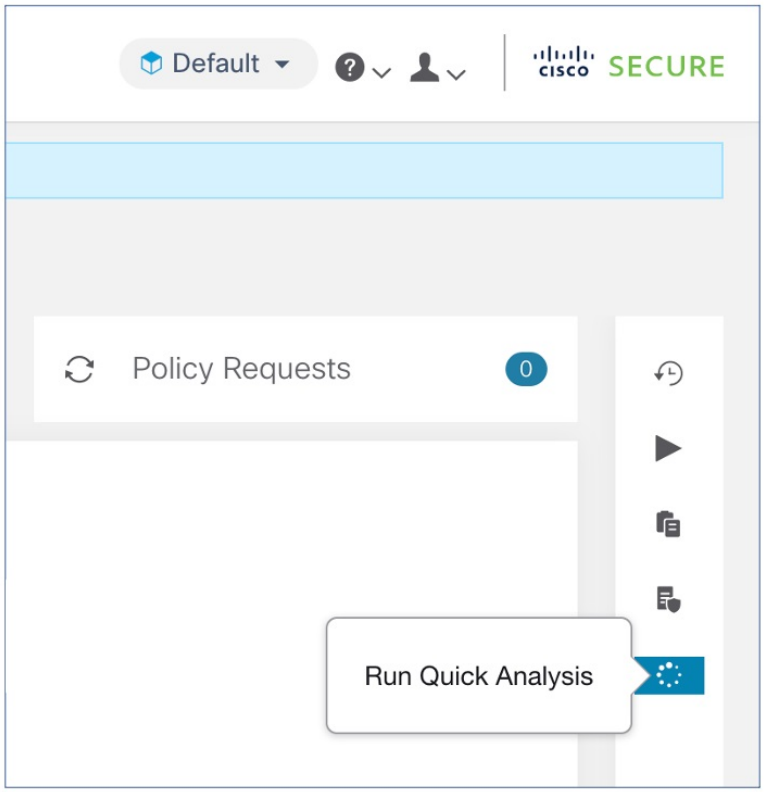
簡易分析により、現在のワークスペース内の全ポリシー、および他のワークスペースからの全関連ポリシーに対する仮想フローをテストできます。簡易分析により、ワークスペースのライブポリシー分析を実行しなくても、さまざまなセキュリティポリシーを使用したデバッグと試験が容易になります。

制約事項

- 簡易分析は、プライマリワークスペースでのみ実行できます。
- 簡易分析は現在、Kubernetes サービスからのフローではサポートされていません。

右側のナビゲーションウィンドウで [簡易分析の実行（Run Quick Analysis）] タブをクリックして、ダイアログを表示します。

図 *316 : [*簡易分析（*Quick Analysis*）*]* タブ



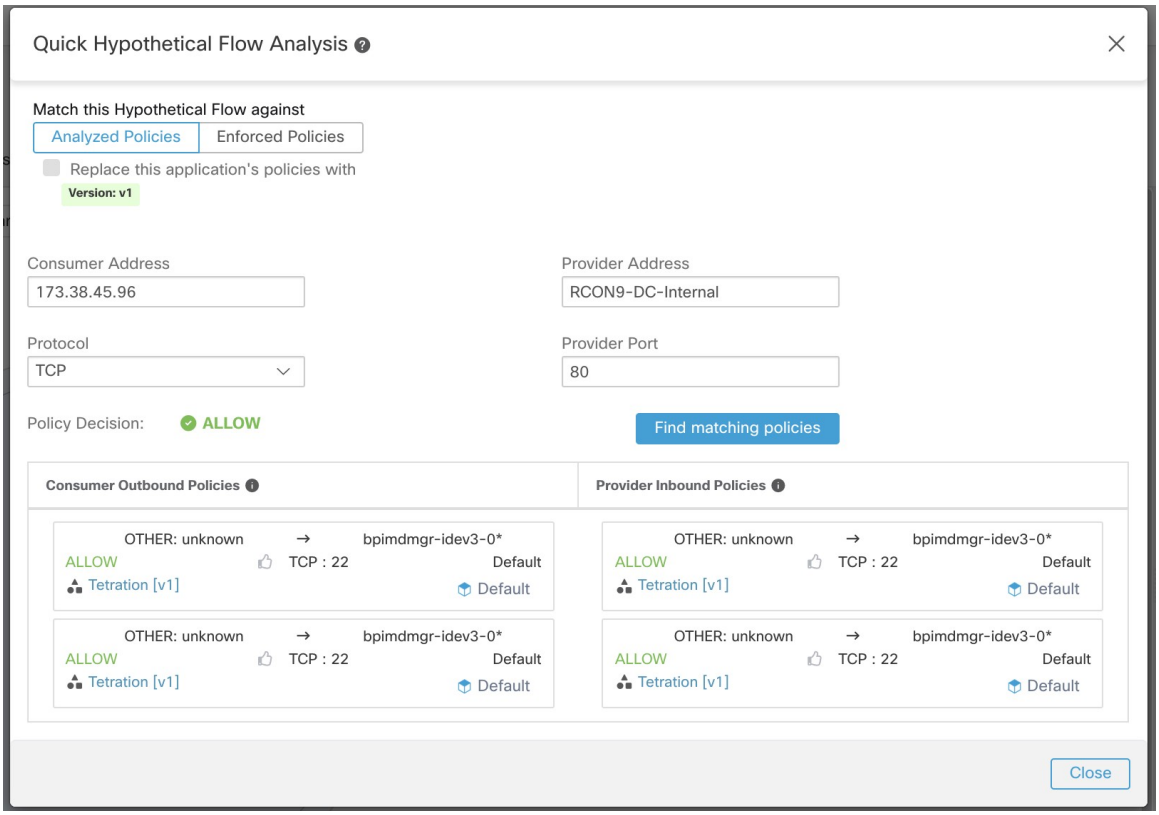
仮想フローのコンシューマ（クライアント）IP、プロバイダー（サーバー）IP、ポート、およびプロトコルを入力し、[一致するポリシーの検索（Find Policy Match）] ボタンをクリックします。

ワークスペースの最新バージョンのポリシー定義と、ライブポリシー分析のために既にプッシュされている関連したワークスペースからの他の全ポリシーを考慮して、仮想フローが許可または拒否されるかどうかを示すポリシー決定が表示されます。

ダイアログの下部に、一致するアウトバウンドポリシーとインバウンドポリシーが個別に、一括で並べ替えられた順序で表示されます。有効なものは、いずれかの側の最初の行のみです。接続を正常に確立するには、コンシューマ側の最上位のアウトバウンドルールとプロバイダー側の最上位のインバウンドルールの両方が ALLOW ルールである必要があります。

他のすべての一致するポリシーを順番に表示すると、特定のポリシーが有効になっていないように見える場合に、ポリシー定義の問題を整理するのに役立つデバッグツールが提供されます。ワークスペースからポリシーを追加、更新、または削除し、すぐに分析を繰り返すことができます。ワークスペースでのライブポリシー分析の実行は必要ありません。

図 *317* : 簡易ポリシー分析



ライブポリシー分析

自動ポリシー検出によって生成された一連のネットワーク セキュリティ ポリシーを確認して承認した後、ポリシーを適用する前に、ライブポリシー分析を使用して、ポリシーがネットワーク上の実際のトラフィックにどのように影響するかを確認する必要があります。

ライブポリシー分析は、次の質問に答えるために役立ちます。

- このワークスペースのポリシーがすぐに適用された場合、この範囲のアプリケーションにはどのような影響がありますか。
- 新しい一連のポリシーを適用していたなら、既知のセキュリティ攻撃/リスクを防ぐことができたでしょうか。

  過去のトラフィックに対して現在のポリシーをテストするポリシー実験の実行（649 ページ）を参照してください。

- ポリシーは期待どおりに機能しますか。

ポリシーがあるすべてのワークスペースに対してポリシー分析を実行する必要があります。特定の範囲内のワークロードが他の範囲内のポリシーの影響を受ける可能性があるため、範囲にポリシーを適用する前に、単一の範囲に対してのみポリシー分析を実行しないでください。特

定の範囲内のトラフィックに影響を与える可能性のあるすべての範囲のポリシーを分析することを検討してください。

次に例を示します。

- ツリー内のこの範囲の上にある範囲で定義されたポリシーが、この範囲内のワークロードに適用される場合があります。
- この範囲内のワークロードが別の範囲内のワークロードと通信する場合、その範囲内のポリシーがこの通信に影響を与える場合があります。その範囲内でポリシー分析が開始されると（または、その範囲でのポリシーの変更後に最新のポリシーが分析されると）、この範囲のポリシー分析結果に影響する可能性があります。

変更によってアプリケーションが破損しないように、ポリシーを変更するたびにポリシー分析を実行する必要があります。

ワークスペースに対してライブポリシー分析を実行することは、ワークスペースの「パブリッシング」と呼ばれることがあります。

ライブポリシー分析の開始

自動ポリシー検出によってワークスペースで生成されたポリシーを確認し、希望どおりであることが確認できたら、ポリシー分析を開始できます。

始める前に

重要 ライブ分析には、同様にライブ分析を実行している他のワークスペースのポリシーの影響が含まれます。いずれかのワークスペースで適用を有効にしていて、そのワークスペースで分析が実行されていない場合、または適用されたポリシーのバージョンが分析されたポリシーのバージョンと同じでない場合、そのワークスペースのライブ分析結果は正確でない可能性があります。

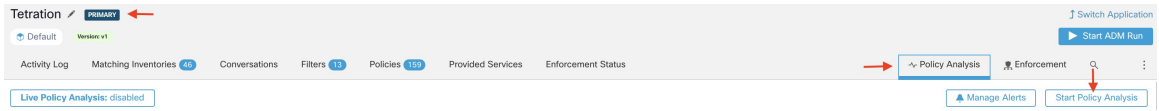
手順

ステップ 1 ヘッダーのワークスペース名の横にある [セカンダリ（Secondary）] の右側の ••• をクリックして、ワークスペースを [プライマリ（Primary）] に切り替えます。

ステップ 2 [ポリシー分析（Policy Analysis）] タブに移動します。

ステップ 3 右側の [ポリシー分析の開始（Start Policy Analysis）] をクリックします。

図 *318*：ポリシー分析の有効化



次のタスク

- 他の範囲のポリシーをこの範囲のワークロードに適用できるため、この範囲の分析結果に影響を与える可能性がある他の範囲のポリシーを同時に分析することを検討してください。例：他の範囲で分析されたポリシーの影響 （644 ページ）を参照してください。
- エスケープされたフローが検出されたときに通知を受け取るには、[アラートの管理（Manage Alerts）] をクリックします。
- このページのツールを使用して、データをフィルタします。使用可能なフィルタ条件を表示するには、フィルタボックスの [(i)] ボタンをクリックします。
- ポリシー分析の開始後にポリシーを追加または変更した場合は、分析に変更内容を含めるために、分析を再起動する必要があります。ポリシーの変更後の、最新のポリシーの分析（651 ページ） を参照してください。

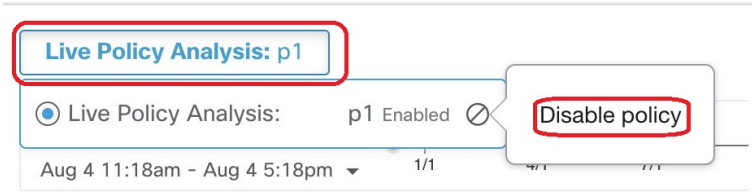
ライブポリシー分析の停止

通常、このワークスペースのポリシーは、分析している他のワークスペースのポリシー分析結果に影響を与える可能性があるため、ポリシーを適用した後でも、ポリシー分析の実行を続行する必要があります。

ライブポリシー分析を停止するには、次の手順を実行します。

[ライブポリシー分析：P<number>（Live Policy Analysis: P<number>）] ボタンをクリックし、[ポリシーの無効化（Disable policy）] をクリックします。

図 *319*：ライブ分析ポリシーの無効化

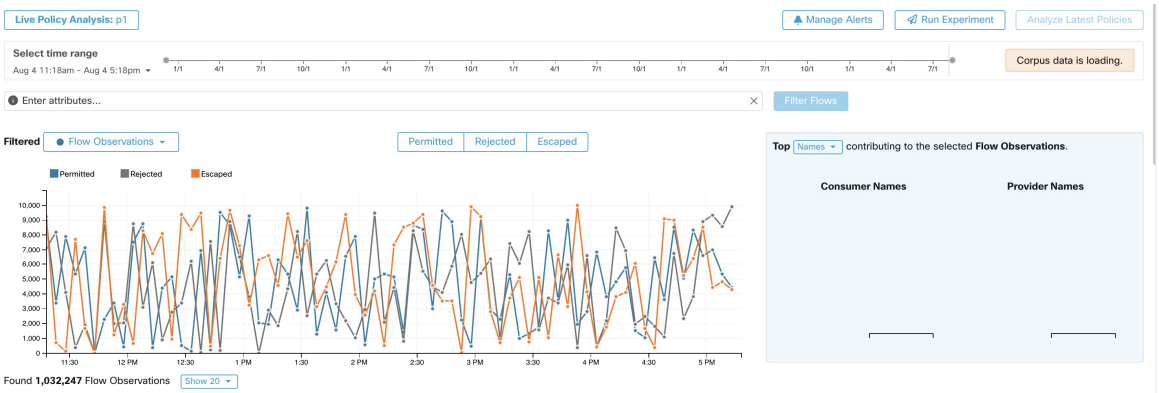


ポリシー分析結果：基本の理解

ポリシー分析中に、ワークスペースに関連付けられた範囲に出入りするすべてのフローと、その範囲内で移動するすべてのフローに、次のいずれかの結果が割り当てられます。

- [許可（Permitted）]：フローはネットワークによって許可され、分析されたポリシーによっても許可されました。
- [エスケープ（Escaped）]：フローはネットワークによって許可されましたが、分析されたポリシーに従ってドロップされました。
- [拒否（Rejected）]：フローは、ネットワークと分析されたポリシーによってドロップされました。

図 *320 :* ポリシー分析ページ



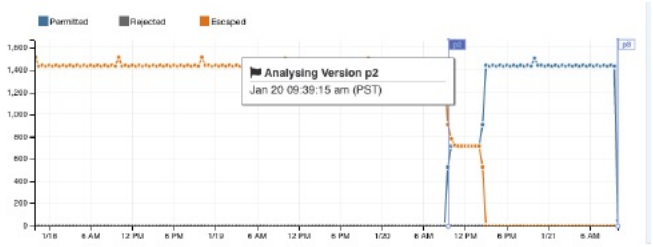
方向性を定めるために、次の点に注目してください。

- ファセットフィルタバーを使用して、このページに表示されるフロー情報をフィルタ処理できます。[フローのフィルタ処理（Filter Flows）] ボタンをクリックすると、それに応じてすべてのチャートが更新されます。
- チャートにカーソルを合わせると、そのタイムスタンプの時点で集約された観測フローの割合が表示されます。
- タイムスタンプをクリックすると、フィルタ処理されたすべてのフローのリストが下の表に表示され、さらに分析できます。
- 時系列チャートの上部にあるタイプを選択または選択解除することで、インタラクションを 3 つの結果タイプのいずれかに制限することができます。
- 右側にある上位N件チャートには、左側の時系列チャートに示されているデータに影響を与えた上位のホスト名、アドレス、ポートなどが示されています。

  時系列チャートをエスケープされたフローだけに制限し、上位 N 件チャートで [ポート（Ports）] を選択すると、エスケープされたフローに影響を与えた上位のポートを表示することができます。

例：他の範囲で分析されたポリシーの影響

次の例では、午後 12 時頃までは、フローが許可されています。その時点で、別の範囲に関連付けられたワークスペースでポリシー分析が開始され、この範囲内のワークロードを持つトラフィックに影響を与えることで、フローがエスケープ済みとしてマークされます（この変更は、このワークスペースで新しく分析されたポリシー変更の結果ではないことがわかっています。この場合は、ラベルフラグが作成されるためです）。



ポリシーなしでの分析

ワークスペースに関連付けられた範囲へのフロー、そのような範囲からのフロー、および範囲内のフローは、分析されている他のワークスペースのポリシーの影響を受ける可能性があります。このワークスペースでライブポリシー分析が有効になっていない場合、フローは、システムに含まれるライブポリシー分析が有効になっている他のワークスペースのフローでマーキングされます。

（注） ライブポリシー分析を実行しているワークスペースがない場合、時系列チャートは空になります。

ポリシー分析の詳細

フローの処置

ポリシーライブ分析では、フローが**許可**、**エスケープ**、または**拒否**のいずれであるかを決定するには、最初にネットワークの観点からフローの**処置**を判定する必要があります。各フローは、Secure Workload エージェントによって提供される信号と観察に基づいて、**ALLOWED**、**DROPPED**、または **PENDING** の処置を受け取ります。フローのパスに沿ったエージェント構成およびフロータイプに基づいた多くのシナリオがあります。

初めに、フロータイプに関係なく、フローのパスに沿ったいずれかのエージェントで、フローが破棄されたことが報告されると、フローは DROPPED の処置を受け取ります。

フローのパスに沿ったエージェントによって破棄が報告されない場合、双方向フローと単方向フローの場合を別々に検討します。双方向フローが観察される場合、送信元、宛先ポート、プロトコル、およびタイミングに基づいて、フローをペア（順方向と逆方向）で調査します。同じことは、単方向フローにはできません。

双方向フローの場合、エージェントがインストールされ、両端でデータプレーンが有効になっている場合、送信元と宛先の両方のエージェントがフローが観察されたことを報告すると、順方向フローはALLOWED の処置を受け取ります。それ以外の場合、順方向フローはPENDING の処置を取得します。送信元または宛先のワークロードのいずれかにエージェントがインストールされていて、両方にはインストールされていない場合、エージェントが **60** 秒の時間枠内に後続の逆方向フローを観察した場合に限り、順方向フローは ALLOWED の処置を受け取ります。それ以外の場合は、PENDING ステータスが順方向フローに割り当てられます。双方向フローの逆方向部分の処置は、送信元と宛先が逆になったことを除いて、同じロジックに従います。たとえば、一方の側にのみエージェントが存在する場合、逆方向フローの処置が

PENDINGかALLOWEDかは、同じロジックに基づく後続の順方向フローの観測とタイミングに依存します。

ファイアウォールがサイレントドロップを実装していると想定していることに注意してください。同じフローで拒否メッセージが送信された場合（例：RST + ACK で TCP SYN を拒否）、逆方向のフローが検出され、以前の順方向のフローが ALLOWED としてマークされます。ただし、拒否メッセージが別のフローで送信される場合（例：ICMP メッセージで TCP SYN を拒否する場合）、順方向フローは PENDING のままになります。

単方向フローの場合、双方向フローの場合と同様に、いずれかのエージェントによってDROPPEDと報告された場合、そのフローはDROPPEDと見なされます。ただし、一致する逆方向フローがないため、両方のエージェントがフローを観察した場合、フローの処置ステータスはPENDINGになります。

違反タイプ

フロー処置は、分析されているポリシーに照らしてチェックされ、最終的な違反の種類が決定されます。

フローの違反タイプは次の通りです。

- **許可**：その処置が ALLOWED または PENDING であり、その決定ポリシーアクションが ALLOW である場合
- **エスケープ**：その処置が ALLOWED であり、その決定ポリシーアクションが DENY である場合
- **拒否**：その処置が DROPPED または PENDING であり、その決定ポリシーアクションが DENY である場合

DROPPED ステータスは、関連するエージェントが明示的に DROPPED ステータスを報告しているフローにのみ割り当てられます。エージェントでドロップが明示的に報告されていない場合、フローは PENDING ステータスを受け取ります。

処置が PENDING の場合は次のようになります。

- ポリシーアクションが DENY の場合、違反タイプは拒否に設定されます。
- ポリシーアクションが ALLOW の場合、違反タイプは許可に設定されます。

双方向フローの場合、フローの順方向部分と逆方向部分のポリシー違反のタイプが一致する場合、ポリシー分析または適用分析ページには1つのタイプのみが表示されます。それ以外の場合は、PERMITTED:REJECTED のように、順方向と逆方向が別々に表示されます。

シナリオの例：

- 送信元側の適用でパケットがドロップされた。
  - この場合、送信元側の Secure Workload 出力エージェントは、フローが DROPPED であると報告します。
- パケットが送信元から出力された。

- 送信元側のエージェントのみが存在する場合、60秒以内にエージェントによって逆方向パケットも観察された場合、フローは出力エージェントによって ALLOWED として報告されます。
- 送信元側と宛先側の両方に可視性のみのエージェントがある場合、入力エージェントがフローが DROPPED であると報告した場合に限り、フローには DROPPED 処置ステータスが与えられます。それ以外の場合、フローは ALLOWED として報告されます。
- 宛先でフローパケットが受信されたが、逆方向トラフィックがない。

  宛先側エージェントがない場合、フローは PENDING ステータスを受け取ります。それ以外の場合は、ALLOWED ステータスが割り当てられます。

フローを調査するための推奨手順

ポリシーの結果を調べるときに特定のフローにドリルダウンする場合は、次の提案とフィルタが役立つ場合があります。

1. まず、エスケープされたフローに焦点を当てます。

   エスケープされたフローには、特別な注意が必要です。これらの実際のフローの処置が、現在分析されているポリシーに基づいて意図されたアクションと異なるためです。これらのポリシーを適用しても、必要なフローがブロックされたり、アプリケーションに悪影響が及んだりすることがないことを調査して確認します。

   [エスケープ（Escaped）] などの違反タイプをクリックします。

   （後で、必要に応じて拒否されたフローと許可されたフローを確認できます）。

   エスケープされたフローは、以下を含みますがこれらに限定されない、さまざまな理由で発生する可能性があります。

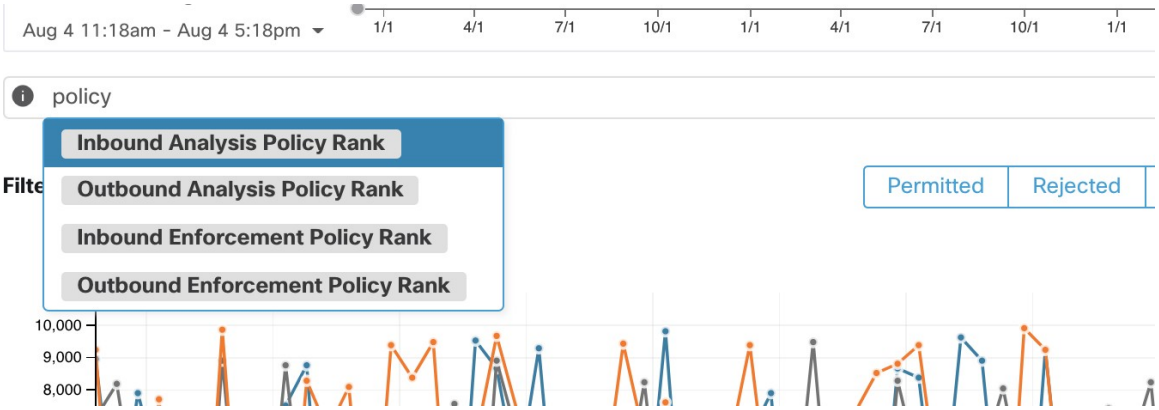
   - 優先順位の高い別のポリシーが有効になっている
   - トラフィックが、ポリシーで扱われるルートとは異なるパスを使用している
   - トラフィックが該当すると予想されるポリシーが、分析されていない（[ポリシー分析（Policy Analysis）] ページでエスケープされたフローを確認している場合）または適用されていない（[適用（Enforcement）] ページでエスケープされたフローを確認している場合）ワークスペース内にある。たとえば、先祖範囲内の場合や、同じ範囲内のセカンダリワークスペース内の場合も該当します。

2. キャッチオールポリシー（インバウンドとアウトバウンド）に一致したフローを特定します。

   特に許可リストポリシーモデルでは、どのフローが Catch-all ポリシーに一致するかを理解することが重要です。これらのフローが正当であっても、これらのフローに対して明示的な許可ポリシーが構成されていない場合、対応するインバウンドまたはアウトバウンド範囲に適切な明示的なポリシーを追加する必要がある場合があります。一方で、疑わしいフローであれば、迅速に特定し、詳細を調査する必要があります。

これらのフローに焦点を当てるために、以下に示すように、注目するのがインバウンド、アウトバウンド、または両側なのかに応じて、**inbound_policy_rank** または **out-bound_policy_rank** の catch-all 値に基づいてフィルタを適用します。

図 *321* : ランクのポリシー分析フィルタリングオプション



3. *Fwd flags does not contain RST, Rev flags does not contain RST* を使用して、RST がある TCP フローを除外します。

   一部のエスケープした TCP フローには、RST フラグが設定されています。これらのフローは、コンシューマまたはプロバイダーのいずれかによってリセットされます。これらは基本的にデータ交換のない確立されていない接続ですが、エージェントがハンドシェイクパケットを認識するため、ALLOWED と報告される場合があります。最初から接続が確立されていないため、現在分析されているポリシーが適用されても影響を受けません。いずれかの側で RST フラグが設定されている TCP フローを除外すると、現在分析されているポリシーによって確立された接続がブロックされる、より意味のある重要なエスケープされたフローに集中できます。

4. ほとんどのトラフィックが IPv4 を使用している場合は、IPv4 フローのみに注目します。

   address type = IPv4, address type != IPv6 を使用してフィルタします。リンクローカルアドレスを除外することにも役立ちます。

5. エスケープされたトラフィックに関連する最も頻繁なホスト名、ポート、アドレス、範囲などを特定することで、次の診断手順で焦点を当てるフローに優先順位を付けます。

   [上位N件（TopN）]機能ペインから、[ホスト名（Hostname）]、[ポート（Ports）]、または[アドレス（Addresses）]を選択します。通常、これらを他のフィルタと組み合わせて、ポリシーを診断するときに特定のタイプのトラフィックにドリルダウンできます。

6. 前の手順で特定したホスト名、ポート、プロトコルなどのフローデータを検索します。

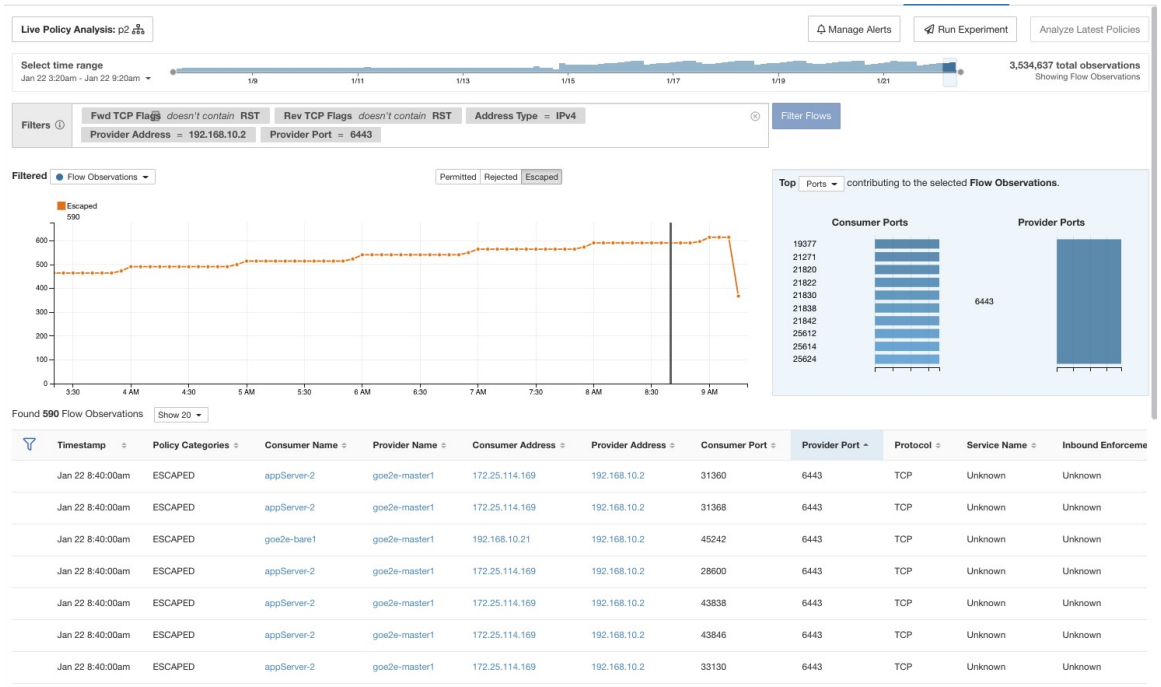
   対象となるフローのホスト名やポートなどに基づいた上位候補について心当たりがあれば、上位N件のクエリウィンドウで取得した値から直接ドリルダウンフィルタを適用するか、フロー検索フィルタバーに関連するフィルタを手動で入力して、フローをドリルダウンすることを選択できます。例：*Consumer Hostname contains {something}, Provider Hostname contains {something}, Provider Port = {some port number}, Protocol = TCP Protocol != ICMP*

7. 個々のフローを確認して迅速に分析します。

   最後に、フローに対応するテーブル行をクリックすることで、特定のフローに注目して、そのポリシーの結果を調べることができます。フローに一致したポリシー、およびコンシューマとプロバイダーの両方のアドレスの範囲に着目してください。ポリシーアクションが意図したアクションと一致しない場合は、コンシューマまたはプロバイダー（またはその両方）の範囲に関連付けられたワークスペースに適切なポリシーを作成して、ポリシーアクションを変更する必要があります。

   次の図は、上記で説明したフィルタリングの一部を使用して、エスケープされたフローを絞り込むワークフローの例を示しています。検索入力は、「-」を範囲クエリに変換することにより、ポート、コンシューマアドレス、プロバイダーアドレスで「,」および「-」をサポートします。

   図 *322* : ポリシー分析診断の例



過去のトラフィックに対して現在のポリシーをテストするポリシー実験の実行

既知の攻撃またはその他の重大な短期トラフィックパターンが過去に発生していて、現在のポリシー（または別のバージョン管理されたポリシーセット）でそのトラフィックがどのように処理されるかを確認する場合は、[実験の実行（Run Experiment）] 機能を使用できます。

始める前に

ヒント　この手順の代わりに、関連する時間範囲を含めて自動ポリシー検出を再度実行し、どのような異なるポリシーが提案されるかを確認することができます。

手順

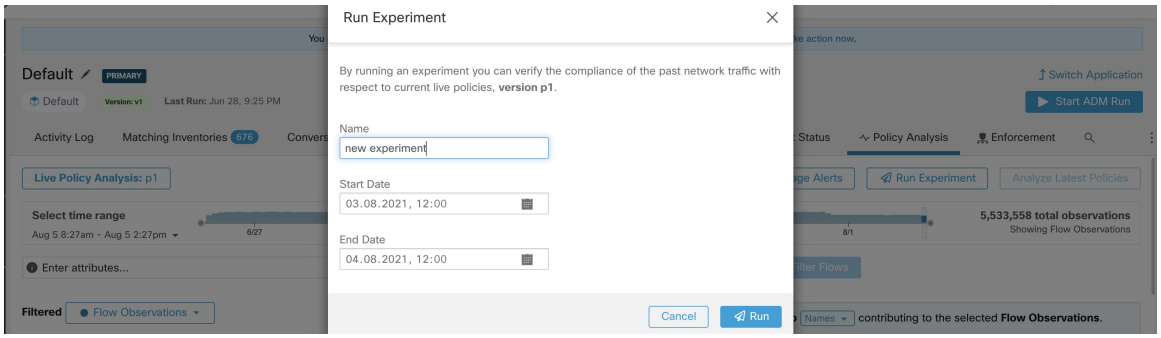
ステップ 1　選択したワークスペースの [ポリシー分析（Policy Analysis）] ページに移動します。

ステップ 2　ページの上部から、テストするポリシーバージョンを選択します。

ステップ 3　ページの右隅にある [実験の実行（Run Experiment）] ボタンをクリックします。

ステップ 4　新しいダイアログで、ポリシー実験の名前と期間を選択します。

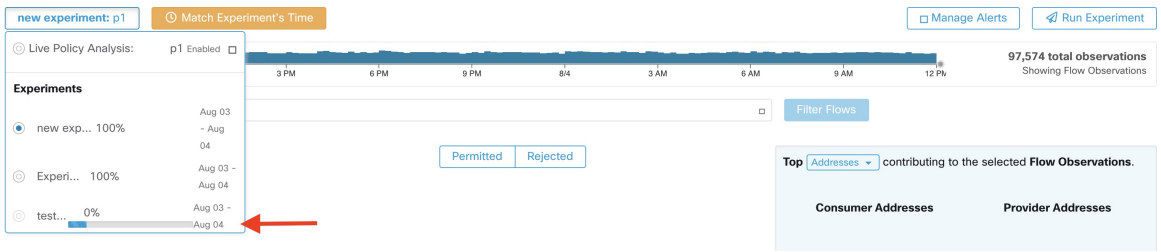
図 *323 : [実験の実行（Run Experiment）] フォーム*



これにより、時間をさかのぼって、選択されたバージョン管理されたポリシーに対して選択された期間内のすべてのフローを再分析する新しいポリシー分析ジョブが開始されます。

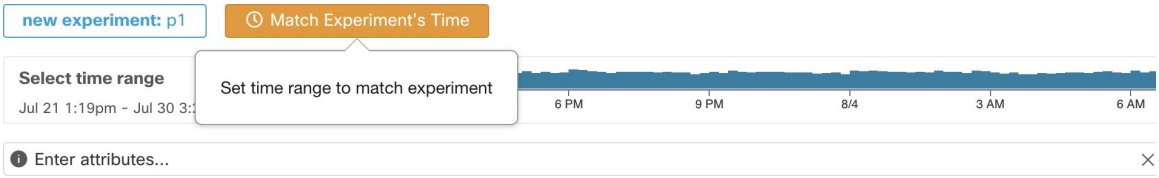
選択した期間によっては、このジョブに数分かかる場合があります。進行状況は、ポリシーセレクタメニューに表示されます。結果を表示する準備ができたら、他のバージョン管理されたポリシーと同様にポリシー実験を選択できるようになり、さまざまなフローカテゴリを示す時系列チャートが選択に応じて更新されます。

図 *324 : 実験ステータスの表示*



（注） ポリシー実験を選択したときにフローが表示されない場合は、時間範囲の不一致が原因である可能性があります。たとえば、チャートの現在の時間範囲が過去1時間なのに、実験期間は過去6時間になっている場合などです。時間範囲を実験期間に合うようにリセットするには、ポリシーセレクタの横にある時計アイコンをクリックします。

図 *325*：時間範囲の適合化



ポリシーの変更後の、最新のポリシーの分析

ポリシー分析では、ワークスペース内のポリシーの変更は自動的に反映されません。変更を加えた後に現在のポリシーのセットを分析する準備ができたら、[最新のポリシーの分析（Analyze Latest Policies）] をクリックして、ポリシー分析に変更を反映させます。

ポリシー分析が最後に開始されてからワークスペース内のポリシーが変更されていない場合、またはポリシー分析が現在有効になっていない場合、[最新のポリシーの分析（Analyze Latest Policies）] ボタンは使用できません。ボタンをクリックできる場合は、まだ分析に含まれていないポリシー変更があります。

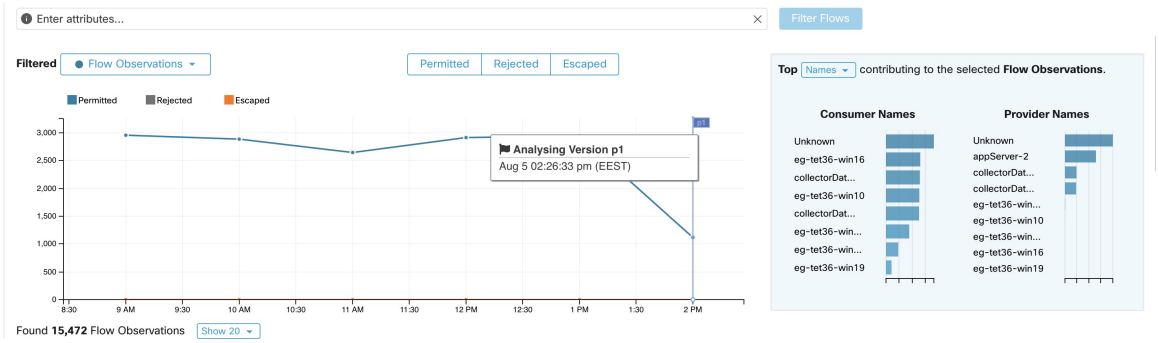
分析されたポリシーバージョンの表示、比較、および管理（652ページ）も参照してください。

ポリシーラベルのフラグ

ポリシー分析の時系列チャートでは、分析が開始された時点と、最新のポリシーとクラスタの変更を反映するために分析が再開始された各時点が、ポリシーラベルのフラグでマークされます。

フラグをクリックすると、そのフラグに関連付けられているポリシーのバージョンが表示されます。

図 *326*：時系列チャートのポリシーラベルのフラグ



ポリシーラベルのフラグをクリックすると、対応するバージョンの[ポリシー（Policies）]ページが開き、そのポリシー分析バージョンによって分析されたポリシーが表示されます。

分析されたポリシーバージョンの表示、比較、および管理

変更後にワークスペース内のポリシーを分析または再分析するたびに、新しい分析バージョン（p*）が作成されます。

バージョン管理の詳細については、ポリシーバージョン（v* および p*）について （673 ページ）を参照してください。

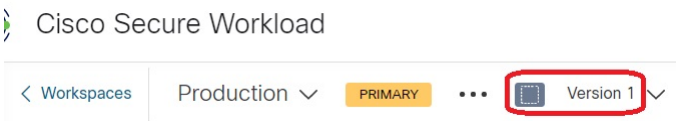
手順

ステップ 1 [防御（Defend）] > [セグメンテーション（Segmentation）] をクリックします。

ステップ 2 関連する範囲とプライマリワークスペースに移動します。

ステップ 3 [ポリシーの管理（Manage Policies）] をクリックします。

ステップ 4 現在表示されているポリシーのバージョンがページの上部に表示されます。



表示されるバージョンには、ポリシー検出バージョン、分析されたポリシーバージョン、または適用されたバージョンがあります。

ステップ 5 次の操作を実行できます。

別のバージョンのポリシーを表示するには、次の手順を実行します。	現在のバージョンをクリックし、別のバージョンを選択します。 バージョンの説明については、ポリシーバージョン（v* および p*）について （673 ページ）を参照してください。 **重要**v* バージョンを選択した場合は、このトピックの代わりに、トピックの最後にある重要な注意事項も含めて検出されたポリシーバージョンの表示、比較、および管理 （570ページ）を参照してください。
分析されたバージョンの詳細を表示するには、次の手順を実行します。	1. ページ上部の現在のバージョンの横にある[バージョン履歴の表示（View Version History）] をクリックします。 2. [公開されたバージョン（Published Versions）] タブをクリックして、分析および適用されたポリシーのバージョンを表示します。 3. バージョンのログエントリを表示するには、バージョン内のリンクをクリックします。 淡い緑色の行は、分析アクティビティを表しています。 明るい緑色の行は、適用アクティビティを表しています。

2つのバージョンを比較して変更点を確認するには、次の手順を実行します。	**1.** [リビジョンの比較（Compare Revisions）] をクリックします。 **2.** 比較するバージョンを選択します。 最新のドラフトバージョン、分析されたバージョン、および適用されたバージョンを比較できます。 **3.** 結果の詳細については、ポリシーバージョンの比較：ポリシーの差分 （676 ページ）を参照してください。
不要なバージョンを削除するには、次の手順を実行します。	バージョンの ⋮ をクリックし、[削除（Delete）] を選択します。 公開されたポリシーバージョン（p*バージョン）は、バージョンがアクティブに分析または適用されていない限り、削除できます。
バージョンをエクスポートするには、次の手順を実行します。	バージョンの ⋮ をクリックし、[エクスポート...（Export...）] を選択します。 ワークスペースのエクスポート （576 ページ）も参照してください。

次のタスク

バージョンの操作が完了したら、ワークスペースページの上部にあるバージョンを、最新の検出されたポリシーバージョン（v*）に変更します。

これにより、検出されたポリシーバージョンの意図しない削除を防ぐことができ、ワークスペースでポリシーを手動で作成することができます。

ポリシー分析のアクティビティログ

すべてのワークスペースユーザーは、ワークスペース履歴のポリシー分析ページで行われた変更に関連するアクティビティログを表示できます（「アクティビティログとバージョン履歴」を参照）。

- ポリシー分析の有効化

  図 *327 :* ポリシー分析の有効化

- ポリシー分析の無効化

  図 *328 :* ポリシー分析の無効化

- ポリシー分析の更新

図 *329* : ポリシー分析の更新

ポリシーの適用

Secure Workload は、次の方法でポリシーを適用できます。

- 個々のワークロードにインストールされたソフトウェアエージェント （23 ページ） ：
  - Linux
  - Windows
  - Kubernetes/OpenShift

  各プラットフォームでのエージェントの動作に関する技術的な詳細については、エージェントによるポリシーの適用 （61 ページ）およびコンテナへの適用 （663 ページ）を参照してください。
- クラウドコネクタ：
  - AWS コネクタ （290 ページ）を介した AWS
  - Azure コネクタ （308 ページ）を介した Azure
- 外部オーケストレータを介した統合ロードバランサ：
  - F5 BIG-IP （200 ページ）
  - Citrix Netscaler （208 ページ）
- Cisco Secure Firewall Management Center （215 ページ） との統合
- サードパーティのインフラストラクチャでの適用のための、サードパーティのオーケストレータへのストリーミング

⚠

注意 ポリシーを適用すると、**影響を受けるホストに新しいファイアウォールルールが挿入され**、関連するホスト上で既存のルールがすべて削除されます。

エージェントの正常性と適用の準備状況の確認

これらのチェックの一部は、ポリシーを適用する前または後に実行できます。

エージェントまたはコネクタの機能を変更するには、権限が必要な場合があります。関連する章の要件と前提条件を参照してください。

ポリシーを適用する予定のないワークロードに対しては、これらのチェックを実行する必要はありません。

確認内容：	詳細情報
エージェントが、適用されたワークスペースに関連付けられた範囲内のすべてのワークロードにインストールされている	[防御（Defend）]>[セグメンテーション（Segmentation）]をクリックし、関連する範囲とワークスペースに移動します。[一致するインベントリ（Matching Inventories）]をクリックし、[IPアドレス（IP Addresses）]をクリックします。 このタブのIPアドレスには通常はエージェントがインストールされていないため、ポリシーを適用するには通常はエージェントをインストールする必要があります。 例外：[IPアドレス（IP Addresses）]タブに表示される次のタイプのインベントリに対しては、適用が行われます。 • クラウドコネクタを使用してポリシーが適用される、クラウドベースのインベントリ（個々のワークロードへのエージェントのインストールはオプションです）。 • エージェントが個々のワークロードポッドにインストールされている場合、Kubernetesアドレスが[IPアドレス（IP Addresses）]リストに表示されます。エージェントがインストールされたKubernetesインベントリが[ポッド（Pods）]タブに表示されます。
インストールされているエージェントのバージョンが最新であり、サポートされている	インストールされているエージェントのバージョンの概要については、[管理（Manage）]>[エージェント（Agents）]をクリックしてから、[分布（Distribution）]をクリックして、[エージェントソフトウェアバージョンの分布（Agent Software Version Distribution）]チャートを確認します。 詳細については、[管理（Manage）]>[エージェント（Agents）]をクリックしてから、[エージェントリスト（Agents List）]をクリックします。
インストールされているエージェントに適用機能がある	[管理（Manage）]>[エージェント（Agents）]をクリックしてから、[適用エージェントに変換（Convert to Enforcement Agent）]をクリックします。 [フィルタ（Filter）]ボックスに、**`Agent Type = Deep Visibility`**と入力します。 ポリシーを適用する必要があるすべてのエージェントを変換します。

確認内容：	詳細情報
すべてのエージェントで適用が有効になっている	（この要件は、エージェントに適用機能があることを確認することや、ワークスペースで適用を有効にすることとは異なります）。 **重要**展開によっては、ワークスペースを適用する前または後にこれを行う必要があります。 「エージェントの適用が有効になっていることの確認」の項を参照してください。
エージェント以外の適用メカニズムで適用が有効になっている	**重要**：ワークスペースでポリシーを適用するまでは、エージェントを使用せずにクラウドコネクタで適用を有効にしないでください。 また、適用を可能にするには、適用をサポートする外部オーケストレータも有効にする必要があります。
エージェント設定プロファイルの [ルールの保持（Preserve Rules）] 設定が、ワークロードプラットフォームに適している	• Kubernetes/OpenShift については、「コンテナへの適用」の項を参照してください。 • 他のプラットフォームについては、「ソフトウェアエージェント」の項の各プラットフォームの情報を参照してください。 ヒント：このドキュメントで「ルールの保持」を検索すると、役立つ情報が見つかります。
（ワークスペースが適用された後に）すべてのエージェントがワークロードに適用可能なポリシーを受信した	「適用されたポリシーがエージェントにプッシュされていることの確認」の項を参照してください。

確認内容：	詳細情報
エージェントが正常である	上記のソースに加えて、次の場所にエージェントの正常性に関する情報があります。 • [管理（Manage）] > [エージェント（Agents）] をクリックしてから、[モニター（Monitor）] をクリックします。 [適用エージェント（Enforcement Agents）] の下の情報を確認します。 • [管理（Manage）] > [エージェント（Agents）] をクリックしてから、[分布（Distribution）] をクリックします。 ページの上部からエージェントタイプを選択します。 • [整理（Organize）] > [範囲とインベントリ（Scopes and Inventory）] をクリックして、対象の特定のワークロードを見つけるためにフィルタし、IP アドレスをクリックします。 [エージェントの正常性（Agent Health）] パネルが含まれている別のブラウザウィンドウで、[ワークロードプロファイル（Workload Profile）] ページが開きます。 詳細については、「ワークロードプロファイル」の項を参照してください。

ポリシーの適用の有効化

⚠

注意 ポリシーを適用すると、既存のファイアウォールルールが削除され、このワークスペースの影響を受ける範囲内のすべてのワークロードに対して新しいファイアウォールルールが作成されます。

ポリシーが正しく機能することを適切に検証していない場合、ポリシーを適用すると、アプリケーションの動作方法が変更され、事業運営が中断される可能性があります。

始める前に

- 最初に、ポリシーを適用する場合は、キャッチオールを [許可（Allow）] に設定することを検討してください。次に、トラフィックをモニターして、キャッチオールルールに一致するものを確認します。キャッチオールルールに一致する必要なトラフィックがない場合は、キャッチオールを [拒否（Deny）] に設定できます。
- 一度に複数の範囲でワークスペースを適用する場合は、分析されたワークスペースのみを適用できます。以下の手順で説明する2番目の方法を使用して単一のワークスペースを適用する場合は、適用する前にワークスペース内のポリシーを分析することを強く推奨しますが、必須ではありません。

「ライブポリシー分析」およびサブトピックを参照してください。

- 単一の範囲を適用するためのウィザードは、複数の範囲を同時に適用するオプションを提供するウィザードよりも詳細です。ポリシー適用ウィザード（661 ページ）の機能が必要な場合は、以下の手順で説明する 2 番目の方法を使用します。
- **重要**：ポリシーが正しいことを確認します。

  ワークスペースのポリシー結果は、他の範囲の適用されたポリシーの影響を受ける場合があります。ワークスペースでポリシーの適用を有効化する前は、[ポリシーの適用（Policy Enforcement）] ページには、別の範囲に関連付けられたワークスペースの適用されたポリシーによって、フローがどのような影響を受けるかが表示されます。たとえば、「実稼働ホストは非実稼働ホストと通信すべきではない（Production hosts should not talk with Non-Production hosts）」という広範なポリシーが親範囲の適用されたワークスペースにあると、子範囲のアプリケーションに属するワークロードのトラフィックに影響を与える場合があります。

  適用チャートに新しい情報が表示されない場合は、正しい時間範囲が選択されているかどうかを確認してください。

  [適用（Enforcement）]ページに表示される情報については、「ライブポリシー分析」およびサブトピックを参照してください（ライブ分析と同じ情報が [ポリシーの適用（Policy Enforcement）] ページにも適用されます）。

  ライブ分析の結果が [適用（Enforcement）] ページの結果と異なる場合は、分析対象の範囲、ポリシーバージョン、および時間範囲が、[適用（Enforcement）]ページで結果を生成するために使用されている範囲、ポリシーバージョン、および時間範囲と同じであることを確認します。

- エージェントが各プラットフォームでポリシーを適用する方法を理解します。参照先：
  - Windows と Linux のワークロードについては、エージェントによるポリシーの適用（61 ページ）およびサブトピックを参照してください。
  - Kubernetes と OpenShift については、以下を参照してください。

    コンテナへの適用 （663 ページ）。
  - ロードバランサについては、以下を参照してください。

    Citrix Netscaler のポリシーの適用 （210 ページ）および

    F5 BIG-IP のポリシーの適用 （203 ページ）。
  - クラウドコネクタを使用して設定されたクラウドベースのワークロードについては、以下を参照してください。
    - AWS インベントリにセグメンテーションポリシーを適用するときのベストプラクティス （304 ページ）およびリンク先のトピック。
    - Azure Inventory にセグメンテーションポリシーを適用するときのベストプラクティス （316 ページ）およびリンク先のトピック。

- ポリシーを適用するには、必要な権限を持っている必要があります。

  範囲に対して適用以上の機能が必要です。範囲で他の機能を持つユーザーは引き続きこのページを表示できますが、新しいポリシーの適用（または無効化）はできません。詳細については、「ロール」を参照してください。

- 関連するすべてのインストールされたエージェントと、クラウドコネクタなどのその他の適用エンドポイントについて、ポリシーを適用する準備ができていることを確認します。

  エージェントの正常性と準備状況チェックのリストについては、エージェントの正常性と適用の準備状況の確認（654ページ）を参照してください（これらのチェックの一部は、適用後まで待機する必要があります。たとえば、ワークスペースで適用を有効にする前ではなく、有効にした後に、クラウドコネクタで適用を有効にする必要があります。インストールされたエージェントについては、通常、ワークスペースを適用する前にエージェント設定で適用を有効にします）。

手順

ステップ1 [防御（Defend）] > [セグメンテーション（Segmentation）] を選択します。

ステップ2 1つの範囲または同時に複数の範囲にポリシーを適用できます。

同時に複数の範囲にポリシーを適用するには、次の手順を実行します。

（このプロセスを使用して適用できるのは、分析済みのワークスペースのみです）。

a) ページの右側にあるキャレット記号をクリックして、[ツール（Tools）]ペインを表示します。



b) [適用の有効化（Enable Enforcement）] をクリックします。

c) [次へ（Next）] をクリックして、ウィザードを開始します。

d) 適用するワークスペースを1つ選択します。

（追加の範囲にワークスペースを適用するオプションは、ウィザードの最後のページにあります）。

e) [次へ（Next）] をクリックします。

f) 適用するそのワークスペースのバージョンを選択し、[次へ（Next）] をクリックします。

g) 別の範囲のポリシーを同時に適用するには、[+別のワークスペースの追加（+ Add Another Workspace）] をクリックし、手順を完了します。

必要に応じて、追加の範囲に対して繰り返します。

h) [同意して適用（Accept and Enforce）] をクリックします。

単一範囲にポリシーを適用するには、次の手順を実行します。

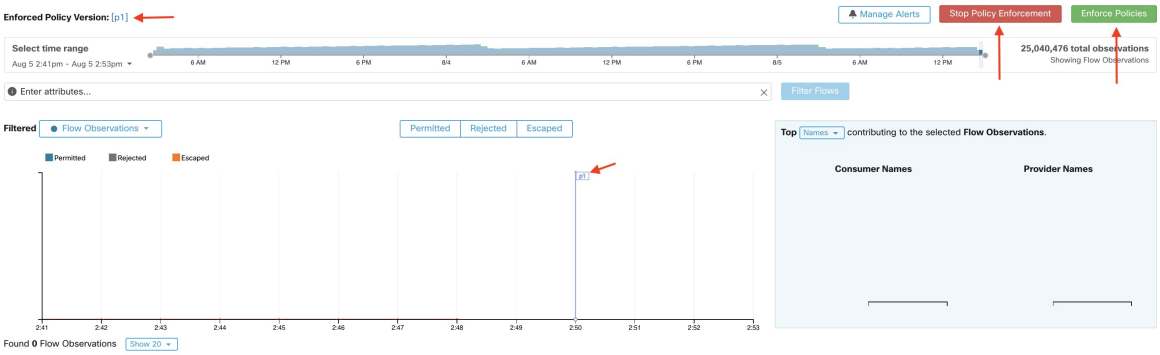
a) ポリシーを適用する範囲のプライマリワークスペースに移動します。
b) [ポリシーの管理（Manage Policies）] をクリックします。
c) [適用（Enforcement）] をクリックします。
d) [ポリシーの適用（Enforce Policies）] をクリックします。
e) ウィザードの手順に従います。

ウィザードの詳細については、ポリシー適用ウィザード （661 ページ）を参照してください。

ステップ 3 ウィザードの最後のページで [同意して適用（Accept and Enforce）] をクリックすると、このワークスペース内のポリシーの影響を受けるアセットに、新しいファイアウォールルールがプッシュされます。

適用時にラベルフラグが作成されます。

図 *330* : 適用が有効化されている *[*ポリシーの適用（ *Policy Enforcement*）*]* ページ



フラグを表示するために、ページを更新する必要がある場合があります。

次のタスク

- 単一のワークスペースにポリシーを適用した場合は、予想される適用結果を得るために、他の範囲のワークスペースにも適用する必要があるかどうかを検討してください。

  たとえば、先祖範囲、またはクロス範囲ポリシーに関連するワークロードを含む範囲のワークスペースにも適用する必要がある場合があります。

- ポリシーを適用するエージェント、クラウドコネクタ、外部オーケストレータに対して適用が有効になるまで、適用は行われません。

- エージェントがインストールされているワークロードの場合は、関連する範囲とインベントリフィルタのエージェント設定でポリシーを適用します。「ソフトウェアエージェントの設定 （87 ページ）」およびサブトピックを参照してください。
- クラウドコネクタを使用して設定されたクラウドベースのワークロードについては、以下を参照してください。
  - AWS インベントリにセグメンテーションポリシーを適用するときのベストプラクティス （304 ページ）およびリンク先のトピック。
  - Azure Inventory にセグメンテーションポリシーを適用するときのベストプラクティス （316 ページ）およびリンク先のトピック。
- Kubernetes と OpenShift については、以下を参照してください。
  - コンテナへの適用 （663 ページ）
  - ソフトウェアエージェントの設定 （87 ページ）
- ロードバランサについては、以下を参照してください。
  - F5 BIG-IP のポリシーの適用 （203 ページ）

    F5 入力コントローラのポリシーの適用 （205 ページ）
  - Citrix Netscaler のポリシーの適用 （210 ページ）

- 適用が期待どおりに機能していることを確認します。適用が予想どおりに機能することの確認 （665 ページ）を参照してください。
- 適用が有効になった後にフローが拒否された場合など、問題が発生した場合に通知されるようにアラートを設定します。

ポリシー適用ウィザード

ワークスペースの [適用（Enforcement）] ページから単一のワークスペースにポリシーを適用する場合、ポリシー適用ウィザードでは次のことができます。

- ワークロードに導入する前にポリシーを確認する。

  これには、先祖範囲から継承されたポリシーが含まれます。
- 確認のためにポリシーの変更をダウンロードする。
- ポリシーバージョンを比較する。
- 適用するワークスペースの分析されたバージョンを選択する。
- 以前のバージョンにポリシーをロールバックする。

ポリシー適用ウィザードのステップ：

1. ポリシー更新の選択

   ワークロードに適用するポリシーのバージョンを選択でき、

   現在適用されているポリシーと選択したバージョンのポリシーの違いが表示されます。

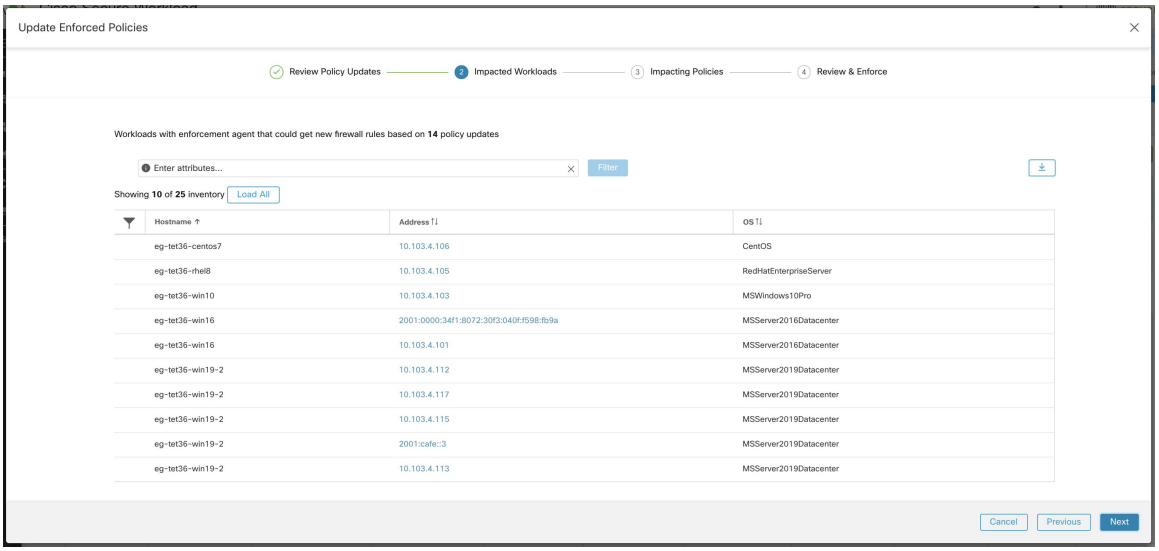
   「ポリシーバージョンの比較：ポリシーの差分」と同様に、ポリシーの変更をフィルタして確認し、CSV としてダウンロードすることができます。

2. 影響を受けるワークロード

   このステップでは、選択したポリシーの変更から生成された新しいファイアウォールルールの影響を受けるワークロードが表示されます。結果は、選択したポリシー変更のコンシューマまたはプロバイダーの結合内に適用エージェントが存在するすべてのワークロードを検索した結果です。

   影響を受ける可能性のあるワークロードの数は、範囲内のワークロードの総数を超えることはありませんが、[エージェント構成インテント（Agent Config Intents）] などの他の要因により、実際に影響を受けるワークロードは少なくなる可能性があります。

   図 *331* : 影響を受けるワークロードのリスト

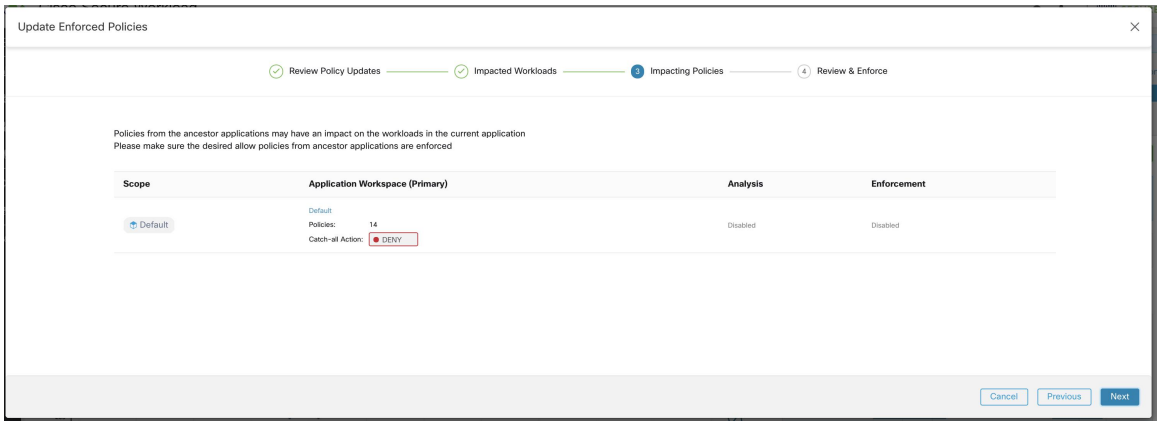


   インベントリ項目の表示、フィルタリング、ダウンロードの詳細については、インベントリ （407 ページ）を参照してください。

3. 影響を与えるポリシー

   先祖ワークスペースのポリシーは、現在のワークスペースのワークロードに影響を与える可能性があるため、先祖ワークスペースから必要な許可ポリシーが適用されていることを確認する必要があります。

図 *332* : 先祖ワークスペースと適用バージョンのリスト

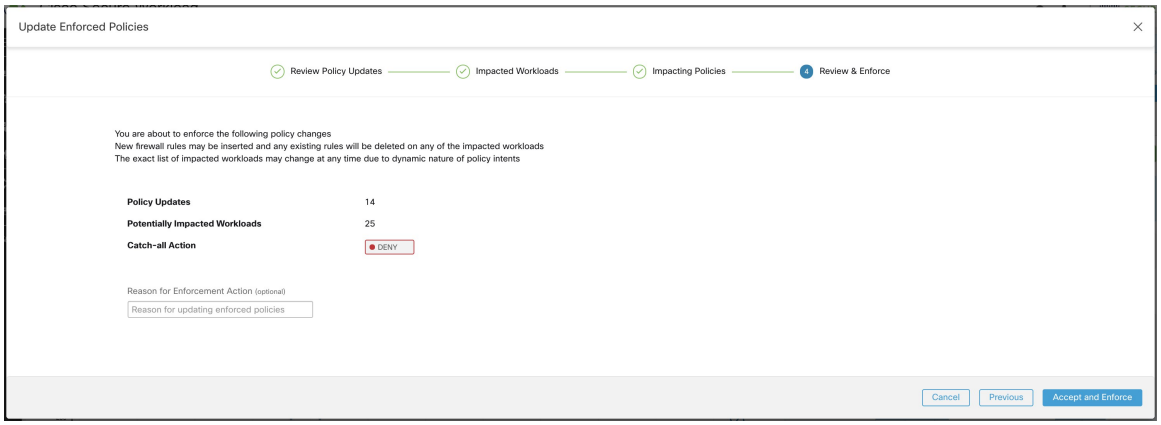


4. 確認と同意

この最後のステップでは、適用されるポリシー変更、影響を受ける可能性のあるワークロードの数、適用されるキャッチオールアクションの概要が示されます。[同意して適用（Accept and Enforce）] をクリックすると、ワークスペースのポリシーを使用して、関連するワークロードに設定される新しいファイアウォールルールが計算されます。

後で参照できるように、新しく適用されたポリシーの名前、説明、およびアクションの理由を提供するオプションがあります。ロールバックの場合、過去のバージョンの名前と説明は変更できないため、理由のみを指定できます。

図 *333* : 概要の確認およびポリシーの変更の適用



コンテナへの適用

Kubernetes と OpenShift によって管理されるコンテナベースのワークロードでセグメンテーションを設定するために必要な手順の概要については、Kubernetes ベースのワークロードに対するマイクロセグメンテーションの設定 （9 ページ）を参照してください。

注目 **Kubernetes や OpenShift ホストで稼働しているエージェントは、既存のルールを保持するように設定する必要があります。**

Kubernetes によって追加された iptables ルールが適用によって干渉されないようにするには、[ルールの保持（Preserve Rules）] オプションが有効になっているプロファイルを使用してエージェントを設定する必要があります。「エージェント設定プロファイルの作成」を参照してください。

コンテナにポリシーを適用すると、Secure Workload は Kubernetes および OpenShift のサービス抽象化をプロバイダーとして使用できるようにします。内部的には、サービス抽象化のポリシーは、プロバイダーポッドとそれらが実行されているノードのルールに変換されます。この変換は、Kubernetes および OpenShift サービスのタイプに依存し、API サーバーから変更を受け取るたびに動的に更新されます。

次の例は、この機能によって実現する柔軟性を示しています。db という名前の NodePort タイプの Kubernetes サービスに対して、ラベル environment = production を持つすべてのホストとポッドからのトラフィックを許可する次のポリシーについて考えます。このサービスは、一連のポッドで TCP ポート 27017 を公開しています。

コンシューマ	プロバイダー	プロトコル/ポート	アクション
environment = production または orchestrator_environment = production	orchestrator_system/service_name = db	TCP 27017	許可

このポリシーにより、次のファイアウォールルールが適用されます。

- environment = production のラベルが付けられたホストとポッドで、サービスが属するクラスタのすべての Kubernetes ノードへの発信接続を許可します。このルールは、Kubernetes によってこのサービスに割り当てられたノードポートを使用します。
- ラベルが environment = production のポッドで、Kubernetes によってこのサービスに割り当てられた ClusterIP への発信接続を許可します。このルールは、サービスによって公開されているポートを使用します（TCP 27017）。
- サービスが属するクラスタの Kubernetes ノードで、プロバイダーポッドへの発信接続を許可します。このルールは、サービスによって公開されているターゲットポートを使用します（TCP 27017）。
- サービスデータベースを提供するポッドでは、すべての Kubernetes ノード、およびコンシューマホストとポッドからのすべての着信接続を許可します。このルールは、サービスによって公開されているターゲットポートを使用します（TCP 27017）。

サービスのタイプ、ポート、および一連のプロバイダーポッドの変更は、Secure Workload のルールジェネレータによってすぐに取得され、生成されたファイアウォールルールを更新するために使用されます。

⚠

注意 **Kubernetes や OpenShift インベントリを含むポリシーは、Kubernetes クラスタの内部操作との競合を避けるために慎重に設計する必要があります。**

Secure Workload によってインポートされた Kubernetes および OpenShift 項目には、Kubernetes クラスタを構成するポッドとサービスが含まれます（例：kube-system 名前空間のポッド）これにより、Kubernetes クラスタ自体を保護するために正確なポリシーを定義できますが、不適切に設計されたポリシーがクラスタの操作に影響を与える可能性もあります。

適用が予想どおりに機能することの確認

エージェントの確認

エージェントの正常性と適用の準備状況の確認 （654 ページ）を参照してください。

エスケープされたフローと拒否されたフローの確認

画面の左側のメニューで、[概要（Overview）] をクリックします。

[セキュリティダッシュボード（Security Dashboard）] ページで、[セグメンテーションコンプライアンススコア（Segmentation Compliance Score）] を確認します。

これが 100 未満の場合は、エスケープまたは拒否されたフローがある可能性があります。どちらもポリシー設定に問題があることを示しています。

詳細は、セグメンテーション コンプライアンス スコア （965 ページ）を参照してください。

これらの状況の調査の詳細については、ポリシー分析結果：基本の理解 （643 ページ）とサブトピックを参照してください（これらのトピックの情報は、[適用（Enforcement）] タブに表示される適用されたポリシーと、[ポリシー分析（Policy Analysis）] タブに表示される分析されたポリシーに適用されます）。

欠落しているポリシーを追加するか、プロトコルやポートの追加などにより既存のポリシーを変更して、必要な正当なトラフィックを許可します。

その後、再適用する前に再分析します。

特定のワークロードの適用されたポリシー（具体的なポリシー）の表示

この手順を使用して、特定のワークロードに適用されているすべてのポリシー（つまり、そのワークロードの具体的なポリシー）を表示します。ワークスペース内のすべてのポリシーがワークスペース内のすべてのワークロードに適用されない場合や、複数のワークスペース内のポリシーが特定のワークロードに適用される場合があるため（たとえば、親または先祖の範囲内の継承されたポリシー）、このビューが役立ちます。

具体的なポリシーは、優先順位順にリストされます。優先順位の影響の詳細については、「ポリシーの優先順位」の項を参照してください。

始める前に

（注） 具体的なポリシーには、適用されたワークスペースのポリシーのみが含まれます。ワークスペースが適用されていない場合、ワークスペースが適用された場合にワークロードに適用されるポリシーは、リストに表示されません。

手順

ステップ 1 [インベントリ（Inventory）] ページまたはワークスペースから、ワークロードの [具体的なポリシー（Concrete Policies）] ページに移動できます。

[範囲とインベントリ（Scopes and Inventory）] ページから移動するには、次の手順を実行します。

a) [整理（Organize）] > [範囲とインベントリ（Scopes and Inventory）] を選択します。
b) 対象のワークロードの IP アドレスを検索してクリックします。

[ワークロードプロファイル（Workload Profile）] が別のタブで開きます。

一般的に、エージェントなしで管理されるクラウドベースのワークロードと、Kubernetes および OpenShift のワークロードを除いて、IP アドレスが [IPアドレス（IP Addresses）] タブには表示されていて、[ワークロード（Workloads）] タブには表示されていない場合は、エージェントがワークロードにインストールされていないことを意味します。そのため、ポリシーを適用できず、具体的なポリシーのリストもありません。

[セグメンテーション（Segmentation）] ページから移動するには、次の手順を実行します。

a) [防御（Defend）] > [セグメンテーション（Segmentation）] を選択します。
b) 範囲をクリックします。
c) プライマリワークスペースをクリックします。
d) [ポリシーの管理（Manage Policies）] をクリックします。
e) [一致するインベントリ（Matching Inventories）] タブをクリックします。
f) 対象のワークロードの IP アドレスを検索してクリックします。
g) 右側に表示されるパネルで、[ワークロードプロファイルの表示（View Workload Profile）] をクリックします。

[ワークロードプロファイル（Workload Profile）] が別のタブで開きます。

ステップ 2 [ワークロードプロファイル（Workload Profile）] ページの左側にあるメニューから、[具体的なポリシー（CONCRETE POLICIES）] をクリックします。

ステップ 3 行をクリックして、詳細を表示します。

詳細については、「[具体的なポリシー（Concrete Policies）] タブ」を参照してください。

ステップ 4 各ポリシーにヒットしたトラフィックの量を表示するには、次の手順を実行します。

a) [すべての統計情報の取得（Fetch All Stats）] をクリックします。

b) 目的の各ポリシーをクリックします。

ステップ 5 Kubernetes または OpenShift のワークロードに関する情報を表示するには、[コンテナポリシー（CONTAINER POLICIES）] をクリックします。

次のタスク

具体的なポリシーのステータスについては、[モニター（Monitor）] > [適用ステータス（Enforcement Status）] を選択します。たとえば、ポリシーがスキップされたかどうかを確認できます。詳細については、「適用ステータス」の項を参照してください。

エージェントの適用が有効になっていることの確認

手順

ステップ 1 [防御（Defend）] > [適用ステータス（Enforcement Status）] をクリックします。

ステップ 2 特定の範囲の適用ステータスのみを表示するには、[範囲でフィルタ（Filter by Scope）] コントロールを切り替えて、範囲を選択します。

ステップ 3 [適用が有効なエージェント（Agent Enforcement Enabled）] チャートを確認します。

いずれかのエージェントが [非適用（Not Enforced）] であることがチャートに示されている場合は、この手順を続行します。

それ以外の場合は、すべてのエージェントで適用が有効になっているため、この手順の残りの部分はスキップします。

ステップ 4 チャートのオレンジの [非適用（Not Enforced）] セクションをクリックすると、チャートの下の表に影響を受けるワークロードが表示されます。

ステップ 5 エージェント設定プロファイルを変更して、これらのワークロードで適用を有効にします。

エージェント設定プロファイルの作成 （90 ページ） を参照してください。

適用されたポリシーがエージェントにプッシュされていることの確認

適用を行うには、各ワークロードに固有のポリシーが、そのワークロードにインストールされているエージェントに正常にプッシュされる必要があります。エージェントがインストールされていない場合でも、クラウドコネクタによって管理されているポリシー適用のステータスは表示されます。

始める前に

少なくとも 1 つの範囲にポリシーを適用します。

手順

ステップ 1 [防御（Defend）] > [適用ステータス（Enforcement Status）] をクリックします。

ステップ 2 特定の範囲の適用ステータスのみを表示するには、[範囲でフィルタ（Filter by Scope）] コントロールを切り替えて、範囲を選択します。

ステップ 3 [エージェントの具体的なポリシー（Agent Concrete Policies）] チャートを確認します。

チャートでいずれかに [スキップ（Skipped）] と表示されている場合は、この手順を続行します。

そうでない場合は、この手順の残りはスキップします。

ステップ 4 この問題の影響を受けるワークロードのリストを表示するには、チャートの赤色の [スキップ（Skipped）] スライスをクリックします。

影響を受けるワークロードが、チャートの下のテーブルに表示されます。

ステップ 5 この問題の原因を確認するには、次の手順を実行します。

検索結果の各ワークロードについて、[具体的なポリシー（Concrete Policies）] 列の [スキップ（Skipped）] の横にある [(i)] ボタンをクリックします。

エラー メッセージ	詳細情報
エージェントに Windows OS がありません（Agent doesn't have Windows OS）	Windows ワークロードにのみ適用されるポリシーの少なくとも 1 つに、Windows OS を実行していないコンシューマやプロバイダーが含まれています。 そのワークロードをそのポリシーから削除してください。
ポリシーの最大数に達しています（Maximum number of policies has been reached）	エージェントのポリシーが多すぎる場合（668 ページ）を参照してください。

次のタスク

（オプション）今後この状況が発生した場合に通知されるようにアラートを設定します。アラート （779 ページ） を参照してください。

エージェントのポリシーが多すぎる場合

すべての適用できる具体的なポリシーを特定のエージェントにプッシュできない場合、ポリシーの最新バージョンはプッシュされません。

背景：各エージェントでサポートされるポリシーの数には制限があります。この制限は、クラウドコネクタを使用して適用されるポリシーにも適用されます。制限（1293ページ）の情報が役立つ場合があります。

始める前に

適用されたポリシーがエージェントにプッシュされていることの確認（667ページ）でエージェントが適用されたすべてのポリシーに対応できないことが示された場合、その問題を解決するにはこの手順を使用します。

手順

ステップ1 影響を受ける範囲のプライマリワークスペースに移動します。

ステップ2 プライマリワークスペースでポリシーを変更します。

コンシューマまたはプロバイダーで、ポリシー数の削減と、IPアドレスの長いリストの削減を試みます。

たとえば、既存のポリシーを統合したり、IPアドレスの大規模なリストではなく、サブネットに基づくポリシーを使用したりします。

クラウドコネクタを使用して適用されるポリシーの場合、プラットフォームによって課される制限を増やすこともできます。クラウドプラットフォームのマニュアルを参照してください。

ステップ3 変更を加えたら、ワークスペースの最新バージョンを適用し、スキップされたポリシーを再度確認します。

ステップ4 この問題が発生している他の範囲に対して、この手順を繰り返します。

適用されたポリシーの変更

新規および変更されたポリシーの適用

適用後にポリシーを変更する必要がある場合は、通常、同じプライマリワークスペースで変更を行います。次に、変更を慎重に確認し、ワークスペースを再度分析して、期待どおりの効果があることを確認します。変更によって目的の効果が得られることを確認したら、ページの右上にある [最新のポリシーの適用（Enforce Latest Policies）] ボタンをクリックします。

適用されたポリシーバージョンの表示、比較、および管理

変更後にワークスペース内のポリシーを適用または再適用するたびに、新しいバージョン（p*）が作成されます。

バージョン管理の詳細については、ポリシーバージョン（v* および p*）について （673 ページ）を参照してください。

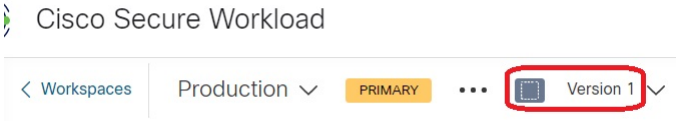
手順

ステップ 1 [防御（Defend）] > [セグメンテーション（Segmentation）] をクリックします。

ステップ 2 関連する範囲とプライマリワークスペースに移動します。

ステップ 3 [ポリシーの管理（Manage Policies）] をクリックします。

ステップ 4 現在表示されているポリシーのバージョンがページの上部に表示されます。



表示されるバージョンには、ポリシー検出バージョン、分析されたポリシーバージョン、または適用されたバージョンがあります。

ステップ 5 次のいずれかを実行します。

別のバージョンのポリシーを表示するには、次の手順を実行します。	現在のバージョンをクリックし、別のバージョンを選択します。 バージョンの説明については、ポリシーバージョン（v* および p*）について （673 ページ）を参照してください。 **重要**v* バージョンを選択した場合は、このトピックの代わりに、トピックの最後にある重要な注意事項も含めて検出されたポリシーバージョンの表示、比較、および管理（570ページ）を参照してください。
分析されたバージョンの詳細を表示するには、次の手順を実行します。	1. ページ上部の現在のバージョンの横にある [バージョン履歴の表示（View Version History）] をクリックします。 2. [公開されたバージョン（Published Versions）] タブをクリックして、分析および適用されたポリシーのバージョンを表示します。 3. バージョンのログエントリを表示するには、バージョン内のリンクをクリックします。 淡い緑色の行は、分析アクティビティを表しています。 明るい緑色の行は、適用アクティビティを表しています。

2つのバージョンを比較して変更点を確認するには、次の手順を実行します。	**1.** [リビジョンの比較（Compare Revisions）] をクリックします。 **2.** 比較するバージョンを選択します。 最新のドラフトバージョン、分析されたバージョン、および適用されたバージョンを比較できます。 **3.** 結果の詳細については、ポリシーバージョンの比較：ポリシーの差分 （676 ページ）を参照してください。
不要なバージョンを削除するには、次の手順を実行します。	バージョンの ⋮ をクリックし、[削除（Delete）] を選択します。 公開されたポリシーバージョン（p*バージョン）は、バージョンがアクティブに分析または適用されていない限り、削除できます。
バージョンをエクスポートするには、次の手順を実行します。	バージョンの ⋮ をクリックし、[エクスポート...（Export...）] を選択します。 ワークスペースのエクスポート （576 ページ）も参照してください。

次のタスク

バージョンの操作が完了したら、ワークスペースページの上部にあるバージョンを、最新の検出されたポリシーバージョン（v*）に変更します。

これにより、検出されたポリシーバージョンの意図しない削除を防ぐことができ、ワークスペースでポリシーを手動で作成することができます。

適用されたポリシーを以前のバージョンに戻す

適用されたポリシーを以前のバージョンにロールバックするには、ポリシーの適用の有効化（657 ページ）で説明されているプロセスのいずれかに従って、適用する以前のバージョンを選択します。

ポリシー適用の無効化

- **複数の範囲のポリシー適用を同時に無効にするには、次の手順を実行します。**

  ポリシーの適用の有効化（657 ページ）で説明されている、複数の範囲で同時にポリシーを適用する手順に従います。ウィザードの [バージョンを選択（Select Version）] ページで、[バージョンを選択してください（Select a version）] をクリックし、[適用の無効化（Disable Enforcement）] を選択します。

- **単一範囲のポリシー適用を無効にするには、次の手順を実行します。**

範囲のプライマリワークスペースの [ポリシー適用（Policy Enforcement）] ページに移動し、赤色の [ポリシー適用の停止（Stop Policy Enforcement）] ボタンをクリックします。これにより、先祖ワークスペースで適用されたポリシーに基づいて、範囲内のアセットに新しいファイアウォールルールが書き込まれます。「x」の付いたラベルフラグが時系列チャートに作成されます。

ポリシー更新の一時停止

⚠

注意 このオプションは、全範囲内のすべてのワークロードのポリシー更新を一時停止します。

この機能には、サイト管理者またはカスタマーサポートの権限が必要です。

全範囲内のすべての適用エンドポイントのルール更新を一時停止するには、次の手順を実行します。

1. [防御（Defend）] > [適用ステータス（Enforcement Status）] を選択します。
2. [ポリシー更新（Policy Updates）] の横にあるステータスをクリックします。
3. 注意を読んで同意します。

図 *334* : ファイアウォールルールが継続的に更新されている場合

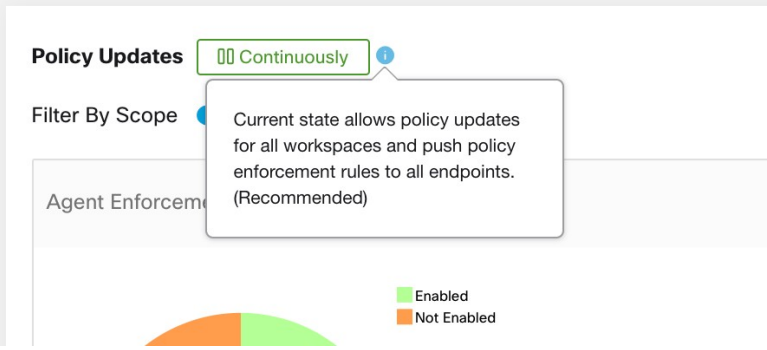


図 *335* : ファイアウォールルールの更新が一時停止されている場合



適用履歴

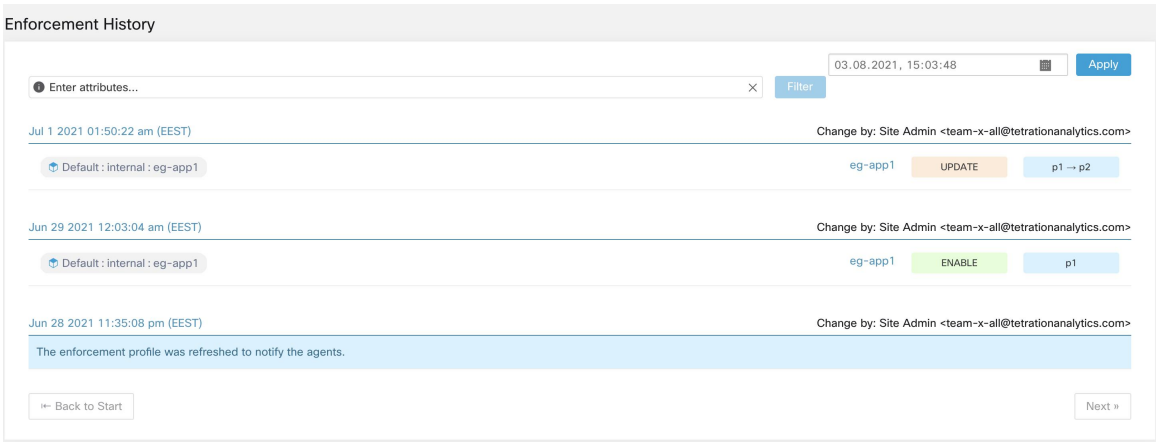
適用履歴には、適用されたワークスペースとそのバージョンのリストに対する変更のリストが表示されます。

適用履歴を表示するには、次の手順を実行します。

1. [セグメンテーション（Segmentation）]ページの右側にあるキャレット記号をクリックして[ツール（Tools）] メニューを展開します。
2. [適用履歴（Enforcement History）] をクリックします。

   各セクションで、イベントについて説明され、変更内容の概要が表示されています。
3. イベントをクリックすると、その時点で適用されていたすべてのポリシーに関する詳細が表示されます。

図 *336 :* 適用履歴ビュー

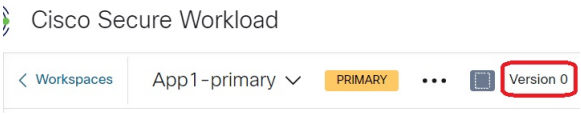


ポリシーバージョン（v* および p*）について

ポリシーバージョンは、ワークスペースバージョンと呼ばれることもあります。

表示されるバージョン

現在作業しているポリシー（およびクラスタ）のバージョンは、ワークスペースページの上部に表示されます。



- V* バージョンは、自動ポリシー検出によって生成されます。

  詳細については、以下を参照してください。

- P* バージョンは、分析されたバージョンや適用されたバージョンです。

  詳細については、以下を参照してください。

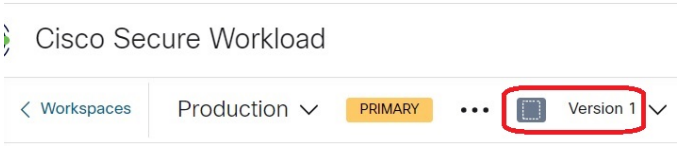
バージョン番号の横に、次のアイコンが表示される場合があります。

表 *30* : バージョンアイコン

	現在分析されているポリシーのバージョンを示します
	現在適用されているポリシーのバージョンを示します
	自動的に検出されたポリシーの最新バージョンを示します
（アイコンなし）	バージョンがそのタイプの最新バージョンではないことを示します

例：

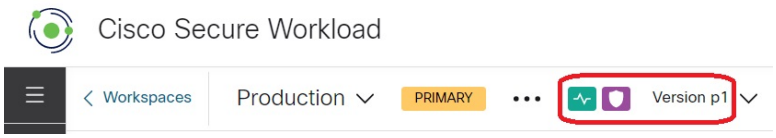
- 表示されているバージョンは、ポリシーの最新の検出されたバージョンです。



- 表示されているバージョンは、現在分析されているポリシーのバージョンです。



- 表示されているバージョンは、現在分析および適用されているポリシーのバージョンです。



ポリシー検出バージョン（v*）

ワークスペースのポリシーを自動的に検出するたびに、バージョン（v*）が増加します。

ポリシーを初めて自動検出すると、バージョン1が生成され、クラスタの（再実行ではない）編集や承認など、実行後のすべての変更もバージョン 1 にグループ化されます。以降、ポリシーを自動的に検出すると、新しいバージョンが生成されます（検出が失敗した場合を除く）。

ポリシーをインポートした場合も、v* バージョンが増加します。

v*バージョンを使用するには、検出されたポリシーバージョンの表示、比較、および管理（570 ページ）を参照してください。

公開されたポリシーバージョン（p*）

ワークスペースの「公開された」ポリシーバージョン（p*）という用語は、次のいずれかを指します。

- 分析されたポリシーのバージョン、または
- 適用されたポリシーのバージョン

これらは、コンテキストに依存する、2 つの独立した類似しているバージョンです。

- 分析のポリシーバージョン：

  ワークスペースでポリシーを分析するたびに、または変更を行った後に [最新のポリシーの分析（Analyze Latest Policies）] をクリックするたびに、そのワークスペースで定義されているすべてのクラスタとポリシーのスナップショットが取得され、分析の「公開された」ポリシーバージョン（p*）の数字が増加します。最新の [ライブポリシー分析（Live Policy Analysis）] バージョンは、プライマリワークスペースの [ポリシー分析（Policy Analysis）] タブのページの左上に表示されます。



- 適用のポリシーバージョン：

  ワークスペースでポリシーの適用を有効にするたびに、または変更を行った後に適用を再度有効にするたびに、適用の「公開された」ポリシーバージョン（p*）が、適用ウィザードで選択した分析されたバージョンの番号になります。そのため、分析されたバージョン 5 を適用すると、たとえば、ワークスペースに初めてポリシーが適用された場合でも、適用されたバージョンもバージョン 5 になります。現在の [適用されたポリシーバージョン（Enforced Policy Version）] は、プライマリワークスペースの [適用（Enforcement）] タブのページの左上に表示されます。



公開された（p*）バージョンの管理

公開されたポリシーバージョンは編集できず、完全な削除のみを実行できます。

（注） 公開されたポリシーバージョン（p*）は、合計100までに制限されます。この制限に達した場合、古いバージョンを削除する必要があります。

p*バージョンを管理および削除するには、分析されたポリシーバージョンの表示、比較、および管理 （652 ページ）または適用されたポリシーバージョンの表示、比較、および管理 （669 ページ）を参照してください。

API を使用して、公開されたバージョンを削除することもできます。

ポリシーバージョンの比較：ポリシーの差分

ポリシーを比較するには、検出されたポリシーバージョンの表示、比較、および管理（570ページ）、分析されたポリシーバージョンの表示、比較、および管理（652ページ）、または適用されたポリシーバージョンの表示、比較、および管理（669ページ）のトピックを参照してください。

ポリシーの変更は、3つのカテゴリ（[絶対（Absolute）]、[デフォルト（Default）]、[キャッチオール（Catch All）]）で表示されます。比較表の説明を以下に示します。

- 同じポリシーに属するさまざまなサービスがグループ化されます
- ポリシーの変更をファセットまたは差分タイプでフィルタします
- ポリシーの変更とサービスはページ分けされます
- フィルタリングされたポリシーの変更を CSV としてダウンロードできます

表 *31* : ファセットフィルタのプロパティ

プロパティ	説明
優先順位	例：100
アクション（Action）	例：ALLOW、DENY
コンシューマ	例：コンシューマクラスタ
プロバイダ（Provider）	例：プロバイダークラスタ
ポート（Port）	例：80
[Protocol]	例：TCP

表 *32 : CSV* 出力列

カラム	説明
ランク	ポリシーのカテゴリ。例：ABSOLUTE、DEFAULT、CATCH_ALL
差分（**Diff**）	変更の差分タイプ。例：ADDED、REMOVED、UNCHANGED
優先順位	例：100
アクション（**Action**）	例：ALLOW、DENY
コンシューマ名（**Consumer Name**）	コンシューマクラスタの名前。
コンシューマ ID（**Consumer ID**）	コンシューマクラスタの ID。
プロバイダー名（**Provider Name**）	プロバイダークラスタの名前。
Provider ID	プロバイダークラスタの ID。
[Protocol]	例：TCP
ポート（**Port**）	例：80

次の図では、ポリシー バージョン p1 と v1 が比較されています。

図 *337 :* ポリシー差分ビュー

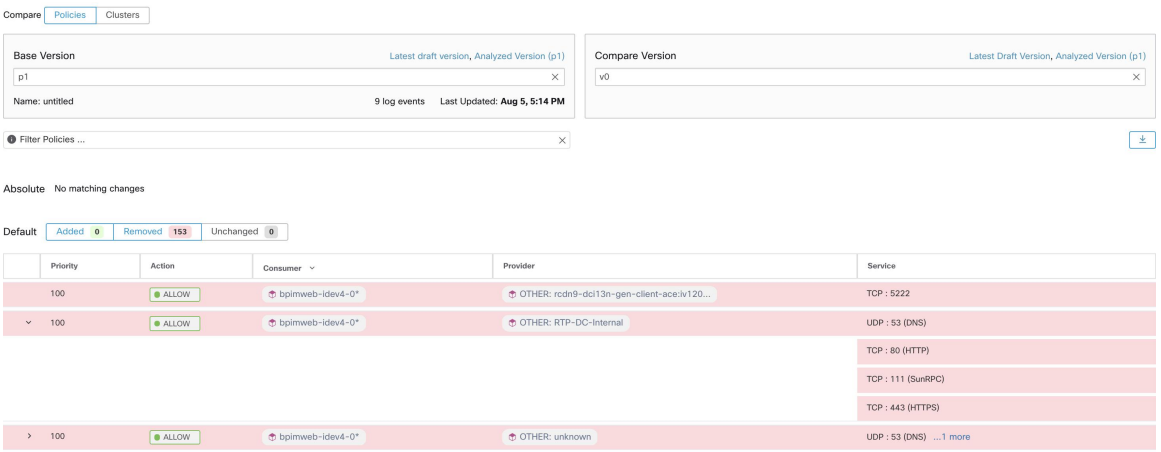


図 *338 :* ポリシー差分ビューのダウンロードボタン

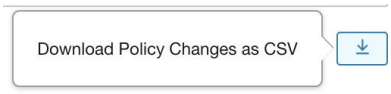


図 *339* : ポリシー差分ビューのフィルタリング

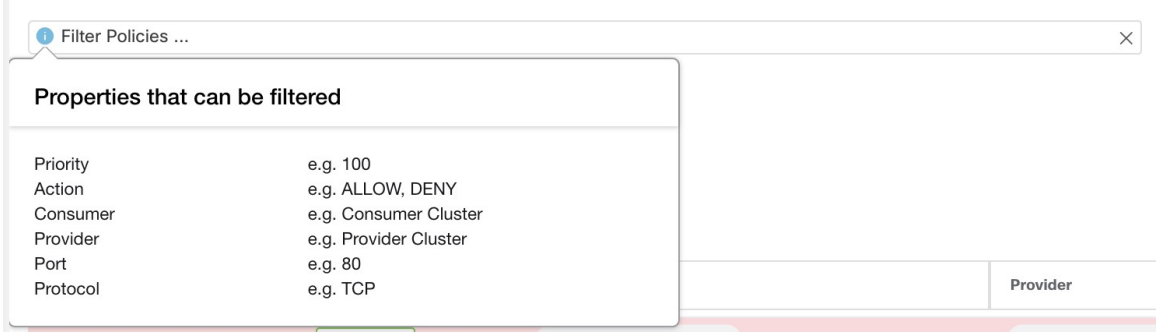


図 *340* : ポリシー差分ビューの差分タイプフィルタ

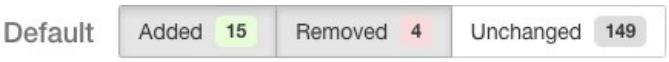


図 *341* : ポリシー差分ビューのグループ化



図 *342* : ポリシー差分ビューの *CSV* 出力

Rank	Diff	Priority	Action	Consumer Name	Consumer ID	Provider Name	Provider ID	Protocol	Port
DEFAULT	ADDED	100	ALLOW	bpimweb-idev4-0*	610bcda7a51e713db909da40	OTHER: rcdn9-dci13n-gen-client-ace:iv120	610bcda7a51e713db909d9f1	TCP	5222
DEFAULT	ADDED	100	ALLOW	bpimweb-idev4-0*	610bcda7a51e713db909da40	OTHER: RTP-DC-Internal	610bcda7a51e713db909d9fe	UDP	53
DEFAULT	ADDED	100	ALLOW	bpimweb-idev4-0*	610bcda7a51e713db909da40	OTHER: RTP-DC-Internal	610bcda7a51e713db909d9fe	TCP	80
DEFAULT	ADDED	100	ALLOW	bpimweb-idev4-0*	610bcda7a51e713db909da40	OTHER: RTP-DC-Internal	610bcda7a51e713db909d9fe	TCP	111
DEFAULT	ADDED	100	ALLOW	bpimweb-idev4-0*	610bcda7a51e713db909da40	OTHER: RTP-DC-Internal	610bcda7a51e713db909d9fe	TCP	443
DEFAULT	ADDED	100	ALLOW	bpimweb-idev4-0*	610bcda7a51e713db909da40	OTHER: unknown	610bcda7a51e713db909da45	UDP	53
DEFAULT	ADDED	100	ALLOW	bpimweb-idev4-0*	610bcda7a51e713db909da40	OTHER: unknown	610bcda7a51e713db909da45	TCP	443
DEFAULT	ADDED	100	ALLOW	bpimweb-idev3-0*	610bcda7a51e713db909da26	OTHER: rcdn9-dci13n-gen-client-ace:iv120	610bcda7a51e713db909d9f1	TCP	5222

ヒント 生成されたクラスタのバージョンの比較：差分ビュー （603 ページ）も参照してください。

アクティビティログとバージョン履歴

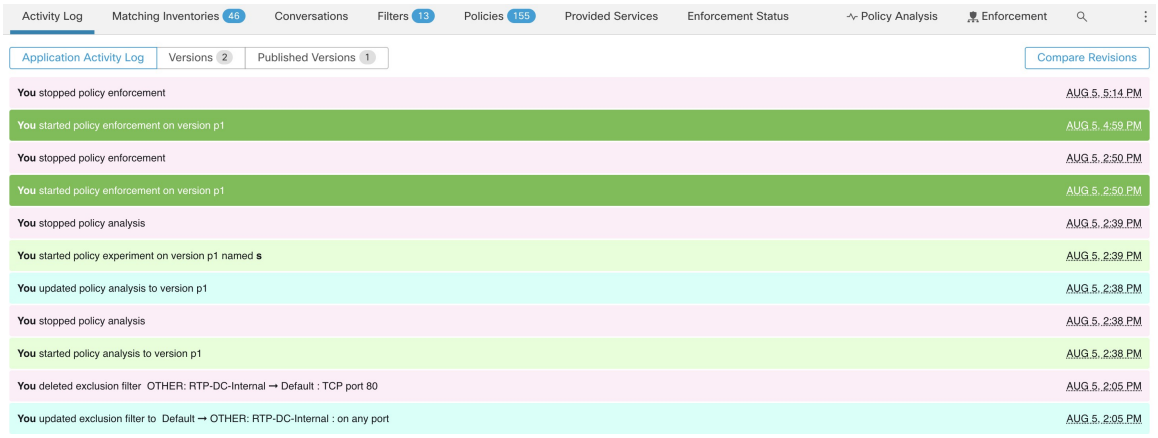
アクティビティログには、すべてのユーザーによる、ワークスペースに適用された変更の履歴が記録されます。表示されるイベントには、ワークロードとクラスタの追加、削除、名前変更、クラスタ間でのワークロードの移動、参考情報のアップロード、自動ポリシー検出の送信と中止などが含まれます。このビューには、それぞれの変更を行ったユーザーが表示されます。

ワークスペースの変更履歴を表示するには、ワークスペースの [アクティビティログ（Activity Log）] リンクをクリックします。

次に例を示します。

1. [防御（Defend）] > [セグメンテーション（Segmentation）] をクリックします。
2. 関連する範囲とワークスペースをクリックします。
3. [アクティビティログの表示（View Activity Logs）] リンクをクリックします。
4. [ワークスペースアクティビティログ（Workspace Activity Log）] タブをクリックします。

図 *343* : このワークスペースのバージョン *v1* に該当するイベントのログ



ページのバージョン関連のタブとオプションについては、次を参照してください。

- ポリシーバージョン（v* および p*）について （673 ページ）
- 検出されたポリシーバージョンの表示、比較、および管理 （570 ページ）
- 分析されたポリシーバージョンの表示、比較、および管理 （652 ページ）
- 適用されたポリシーバージョンの表示、比較、および管理 （669 ページ）

古いポリシーバージョンの自動削除

6ヵ月間アクセスされていないワークスペースバージョンと、過去30日間アクセスされていないポリシー実験は、毎週自動的に削除されます。

カンバセーション

カンバセーションとは、特定のポートの1つのホストによって提供され、別のホストによって消費されるサービスとして定義されます。このようなカンバセーションは、異なるタイミングの多数のフローから出現します。自動ポリシー検出は、そのようなすべてのフローを取得し、一時/エフェメラルポートを無視し、それらを重複排除してカンバセーショングラフを生成します。サーバー（プロバイダー）ポートNでのホストAとホストBの間の特定のカンバセー

ションでは、自動ポリシー検出が実行された時間枠内で、ポート N での A から B へのフローが少なくとも 1 回観測されています。

フローデータを使用すると、自動ポリシー検出中に生成されたクラスタを評価しながら、どのフローがどのプロセスに関連付けられているかをより深く理解できるようになります。

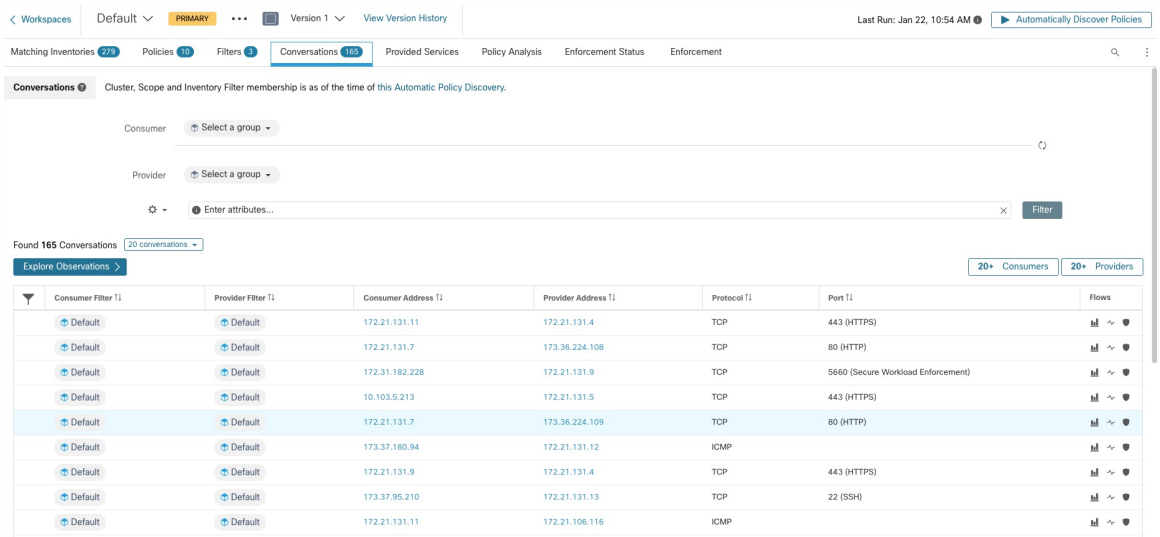
さらに、エージェントによって収集された情報は、未使用の L4 ポートを可視化します。未使用のポートとは、自動ポリシー検出に選択された間隔にわたり通信が確認されなかったポートです。この情報を使用して、これらのポートでの通信に関するポリシーを開いたり、未使用のポートにバインドされているアプリケーションを閉じたりして、ワークロードの攻撃対象領域を減らすことができます。

クライアント/サーバーの分類は、自動ポリシー検出のカンバセーションビューに影響することに注意してください。これは、アグリゲーションでどのポートがドロップされる（エフェメラルと見なされる）かを示します。「クライアントサーバーの分類」を参照してください。

会話テーブルビュー

会話テーブルビューには、コンシューマポートが削除され、レコードが常に1つしかない自動ポリシー検出の期間からの集約されたフローを表示する簡単な方法が示されます。ポリシーはフィルタ間を移動し、会話は IP アドレス間を移動します。

図 *344* : 会話テーブルビュー

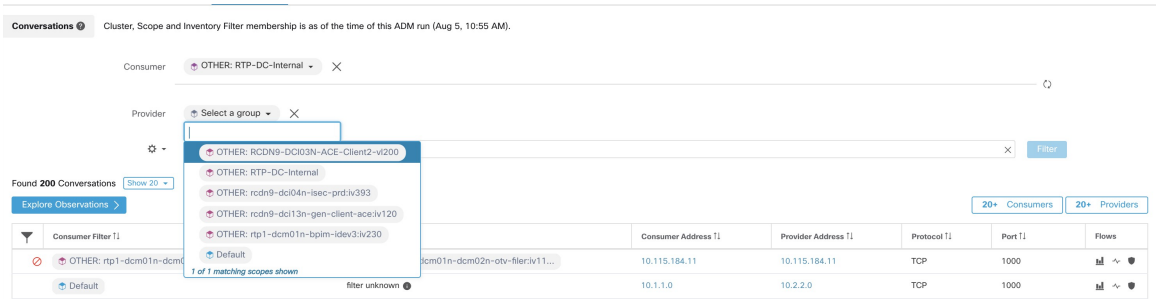


コンシューマまたはプロバイダーの選択

コンシューマとプロバイダーは、下の例に示すように、インベントリフィルタ、範囲、およびクラスタを選択できる、先行入力ドロップダウンセレクタによって選択できます。選択したコンシューマとプロバイダー間のすべてのカンバセーションが表示されます。注：既存のフィルタを削除するには、「x」アイコンをクリックします（フィルタを削除しても機能しない場合があります）。

デフォルトでは、コンシューマとプロバイダーは、自動ポリシー検出中に、IPアドレスがメンバーになっているすべてのインベントリフィルタと照合されます。たとえば、「ルート範囲」を検索すると（一部の IP はより具体的な範囲に一致する可能性がありますが）、すべてのカンバセーションに一致します。より具体的な一致を実行するには、ファセットフィルタ入力の左側にある設定ドロップダウンから[範囲フィルタ処理をIPのベストマッチに制限する（Restrict scope filtering to an IP's best match）] を選択します。

図 *345 :* コンシューマまたはプロバイダーの選択



カンバセーションフィルタ

図 *346 :* カンバセーションフィルタ



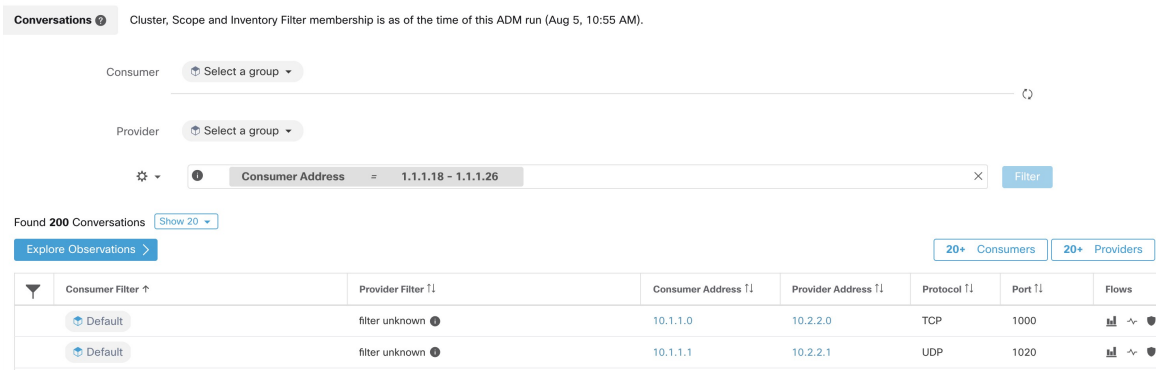
ここで、検索結果を絞り込むためのフィルタを定義します。[フィルタ（Filters）] という文字の横にある[?]アイコンをクリックすると、ディメンションの候補がすべて表示されます。ユーザーラベルデータについては、これらの列も適切な間隔で使用できます。この入力では、キーワード and、or、not、括弧もサポートされており、これらを使用してより複雑なフィルタ条件を表現できます。たとえば、IP 1.1.1.1 と 2.2.2.2 間の方向に依存しないフィルタは次のように記述できます。

Consumer Address = 1.1.1.1 and Provider Address = 2.2.2.2 or Consumer Address = 2.2.2.2 and Provider Address = 1.1.1.1 And to additionally filter on Protocol = TCP:

(Consumer Address = 1.1.1.1 and Provider Address = 2.2.2.2 or Consumer Address = 2.2.2.2 and Provider Address = 1.1.1.1) and Protocol = TCP

フィルタ入力機能は、「-」を範囲クエリに変換することで、ポート、コンシューマアドレス、プロバイダーアドレスの「,」と「-」もサポートします。以下に有効なフィルタの例を示します。

図 *347* : コンシューマアドレスの範囲クエリをサポートするフィルタ入力機能



使用可能なフィルタ：

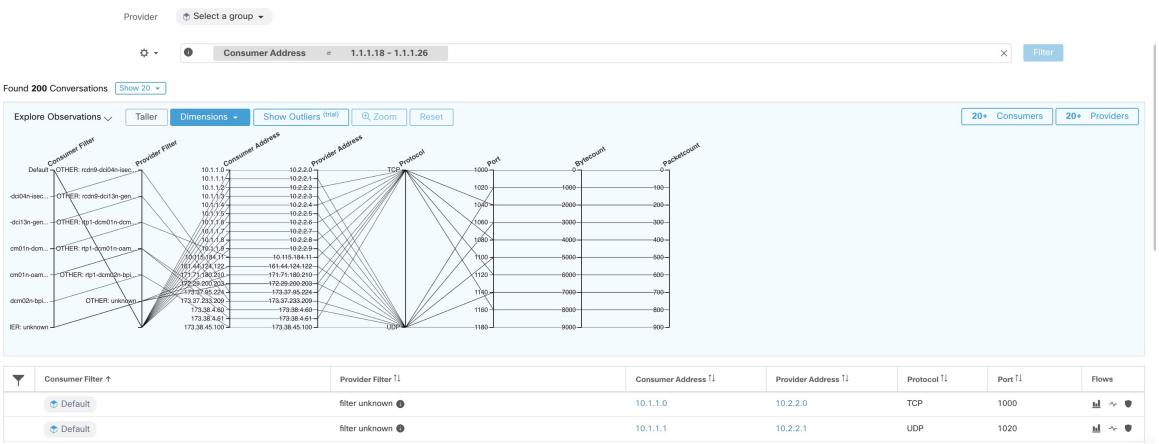
フィルタ	説明
コンシューマアドレス（**Consumer Address**）	CIDR 表記を使用してサブネットまたは IP アドレスを入力します（例：10.11.12.0/24）。コンシューマアドレスが入力した IP アドレスやサブネットと一致するフロー観測データを照合します。
プロバイダーアドレス（**Provider Address**）	CIDR 表記を使用してサブネットまたは IP アドレスを入力します（例：10.11.12.0/24）。プロバイダーアドレスが入力した IP アドレスやサブネットと一致するフロー観測データを照合します。
ポート（**Port**）	ポートが入力したポートと一致する通信フローを照合します。
プロトコル（**Protocol**）	プロトコルタイプ（TCP、UDP、ICMP）を指定して通信フローの観測データをフィルタリングします。
アドレス タイプ（**Address Type**）	アドレス タイプ（IPv4、IPv6、DHCPv4）を指定して対話フローの観測データをフィルタリングします。
信頼性（**Confidence**）	フローの方向の信頼性を示します。入力可能値：[高い（High）]、[非常に高い（Very High）]、[中程度（Moderate）]。
除外対象?（**Excluded?**）	除外フィルタまたは承認済みポリシーを指定して除外対象の対話データを照合します。

フィルタ	説明
除外基準（**Excluded By**）	特定のフィルタを指定して除外対象の対話データを照合します。入力可能値：[除外フィルタ（Exclustion Filter）]、[ポリシー（Policy）]。

観測結果の確認

[観察結果の参照（Explore Observations）] ボタンをクリックすると、チャートビューが有効になり、[平行座標（Parallel Coordinates）] チャートを介して高次元データをすばやく探索できます。最初は少し難しいかもしれませんが、関心のある次元のみを有効にする場合（[次元（Dimensions）] ドロップダウンの項目のチェックを外す) や、次元の順序を並べ替える場合、このチャートは非常に便利です。このチャートの 1 本の線は 1 つの観測値を表し、その線がさまざまな軸と交差する場所は、その次元での観測値を示しています。より明確にするには、チャートの下にある観測結果のリストにカーソルを合わせます。これにより、チャート内の観測結果を示す線が強調表示されます。

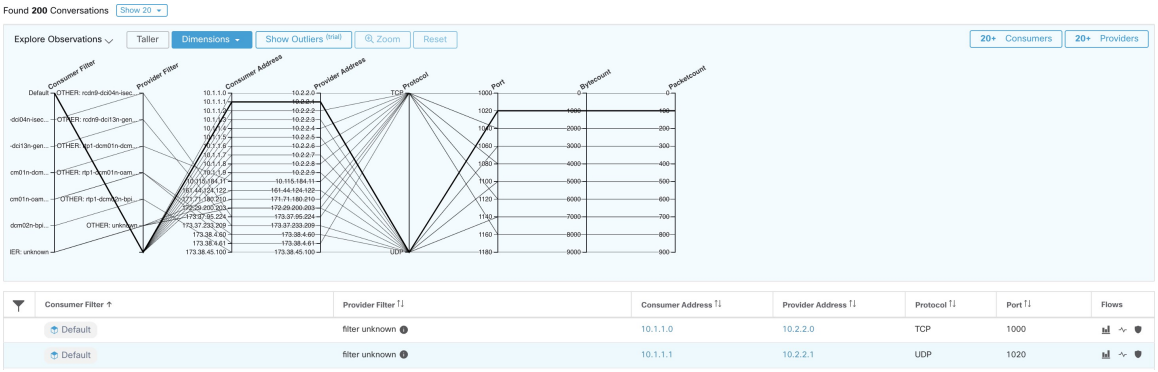
図 *348 :* 観測結果の確認



ホバーによるカンバセーション観測

カンバセーションデータの高次元の性質により、このチャートはデフォルトではかなり幅が広く、チャート全体を表示するには右にスクロールする必要があります。このため、関心のある次元以外はすべて無効にすると便利です。[カンバセーションの詳細確認（Explore Conversations）] のホバー状態は、各カンバセーションをテーブルリストビューにマッピング（ホバー）するために提供されます。

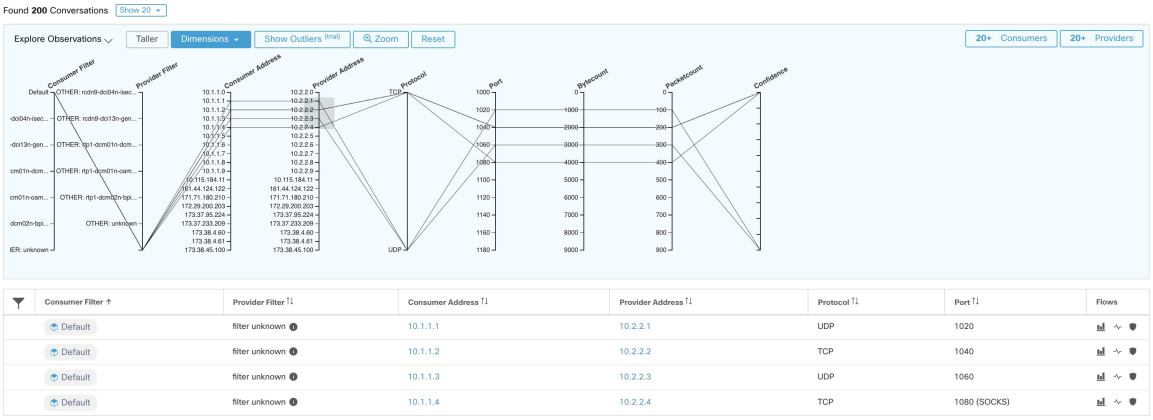
図 *349 :* ホバーによるカンバセーション観測



フィルタリング

いずれかの軸に沿ってカーソルをドラッグすると、選択範囲が作成され、その選択範囲に一致する観測データのみが表示されるようになります。軸を再度クリックすると、いつでも選択範囲を解除できます。一度に任意の数の軸を選択できます。監視データのリストが更新され、選択した会話のみが表示されます。

図 *350 : Filtering*

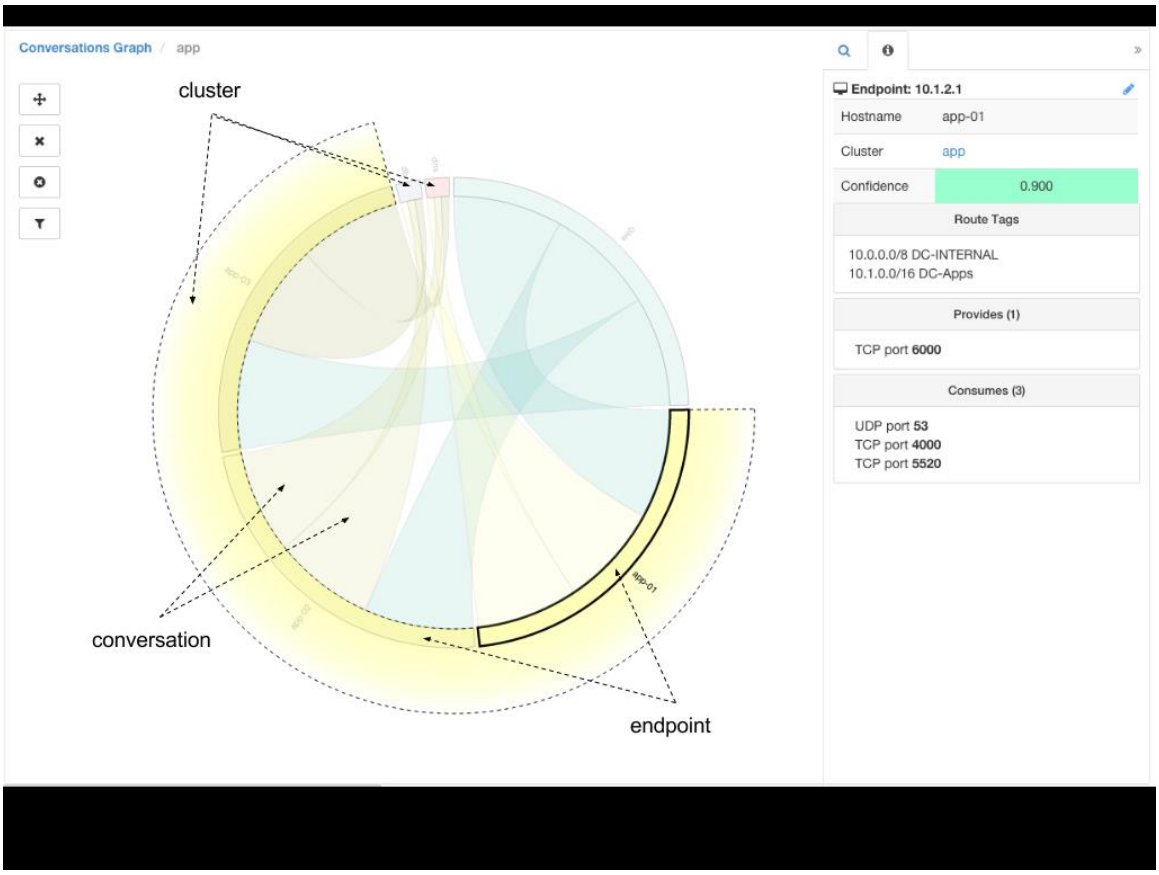


会話チャートビュー

会話チャートビューは、パーティション/クラスタ/ポリシーの代わりにクラスタ/ワークロード/会話に焦点を当てていることを除いて、[ポリシービュー（Policy View）] ページと非常に似たルックアンドフィールを備えています。下の図に示すように、上位レベルの外側の弧はクラスタを表し、展開してメンバーのホスト/ワークロードを内側の弧として表示できます。コードは会話や接続を表します。

会話ビューのコントロールとサイドパネルは、ポリシービューと同様に動作しますが、サイドパネルの情報には、使用または提供されたサービスなどの選択したワークロードに関する詳細情報、および親クラスタへのリンクとプロセス情報（可能な場合）も表示される点が異なります。

図 *351* : 会話チャートビュー



カンバセーションにおける上位のコンシューマとプロバイダー

選択したフィルタを反映したカンバセーションに基づく上位のコンシューマまたはプロバイダーの数は、カンバセーションテーブルの上部にある2つのボタンから確認できます。それぞれのボタンをクリックすると、カンバセーション数の列と、各コンシューマおよびプロバイダーのアドレス、ホスト名、およびその他のユーザー注釈の列を含む表を含むダイアログが表示されます。

図 *352* : カンバセーションテーブルの上部



図 *353* : 上位のコンシューマモーダル

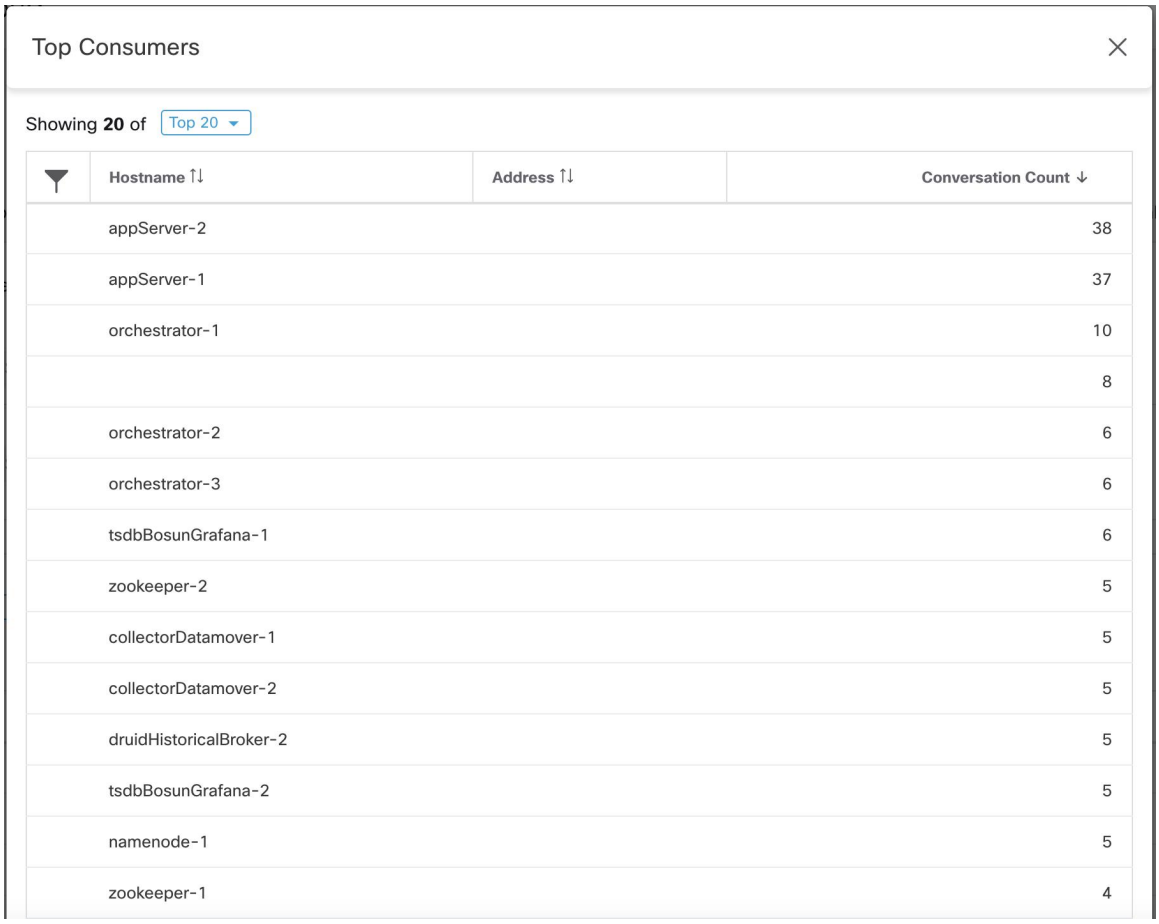
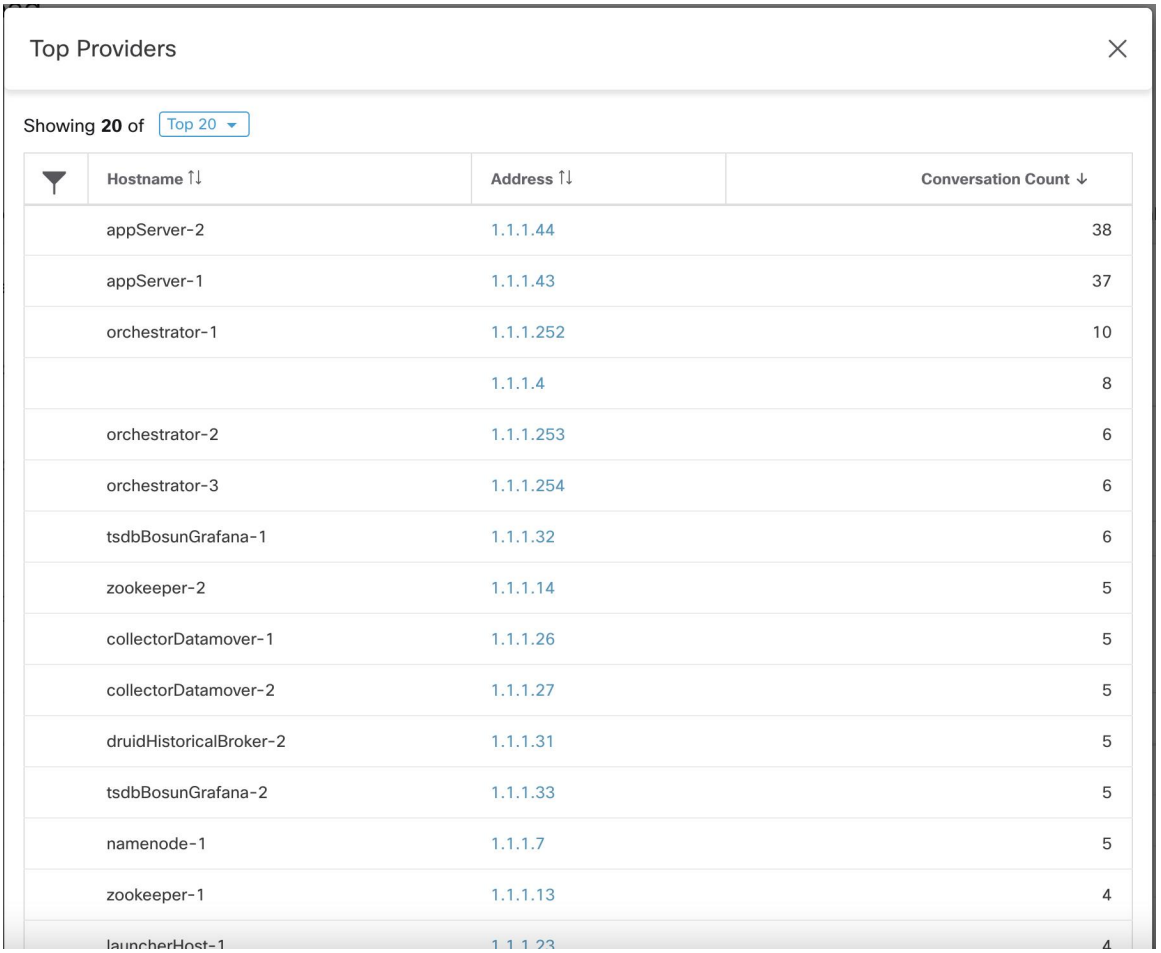


図 *354* : 上位のプロバイダーモーダル



自動ポリシー検出用の自動ロードバランサ設定（F5 のみ）

重要 **これは実験段階の機能です**

この機能と API は**アルファ版**であり、今後のリリースで変更および拡張される可能性があります。

自動ポリシー検出は、外部オーケストレータに接続されたロードバランサの設定からポリシーを生成します。設定からポリシーを生成すると、フローデータへの依存が最小限に抑えられ、検出されるクラスタとポリシーの精度が向上します。

このトラフィックを許可するポリシーを生成するためのロードバランサへのフローの報告は、コネクタに依存しています。

用語

VIP 仮想 IP：クライアントがサービス宛てのトラフィックを送信する IP。

SNIP SNAT IP：トラフィックをバックエンドホストに送信するためにロードバランサによって使用される IP。

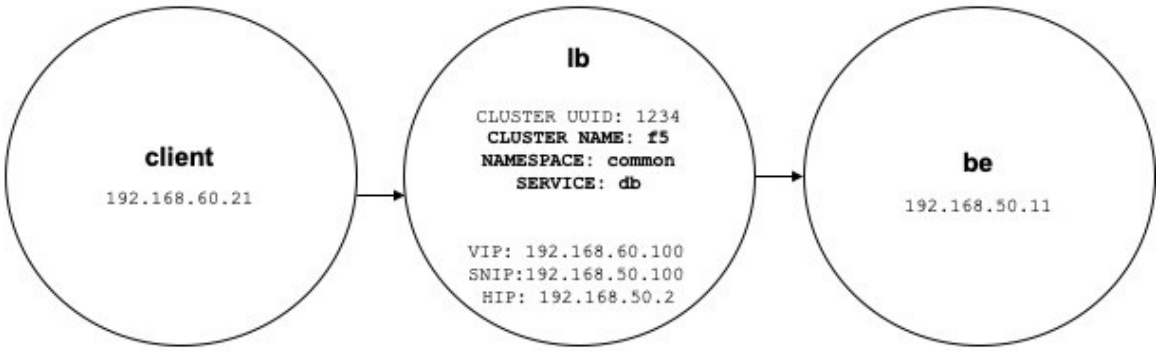
BE バックエンドエンドポイント：バックエンドホストの IP。

HIP ヘルスチェック IP：ヘルスチェックトラフィックをバックエンドホストに送信するためにロードバランサが使用するソース IP。

（注） HIPは、自動マップモードのSNIPと同じです。ただし、SNATプールが設定されている場合、HIP と SNIP は異なる場合があります。

展開（Deployment）

図 *355 :* 展開（*Deployment*）



次のように、ロードバランサの VIP、SNIP、および HIP が *lb* 範囲の一部であり、BE が *be* 範囲の一部である展開を検討してください。範囲は次のように作成されます

- client

  クライアントの範囲には、ロードバランサと通信するクライアントが含まれます。上記の例では、クライアントの範囲のクエリは次のようになります。

  ```
  address eq 192.168.60.21 or address eq 192.168.60.22
  ```

- lb

  F5 外部オーケストレータは、ロードバランサによって使用される VIP、SNIP、HIP、および BE にラベルを付けます。これらのラベルを使用して範囲クエリを構築できます。ここで、*orchestrator_system/service_name* は、サービスの VIP、*orches-*

trator_system/service_startpoint SNIP、*orchestrator_system/service_healthcheck_startpoint* HIP の選択に使用されます。上記の例では、サービス db の VIP、SNIP、および HIP を含む範囲クエリは次のとおりです。

```
user_orchestrator_system/cluster_id eq 1234 and
(user_orchestrator_system/service_name eq db or
user_orchestrator_system/service_startpoint eq db or
user_orchestrator_system/service_healthcheck_startpoint eq db)
```



（注） SNIP と VIP は同じ範囲の一部である必要があります。

- be

  user_orchestrator_system/service_endpoint は、サービスの BE を選択します。上記の例では、サービス db の BE を含む範囲クエリは次のとおりです。

  ```
  user_orchestrator_system/cluster_id eq 1234 and
  user_orchestrator_system/service_endpoint eq db
  ```

クラスタ

各サービスは、最大4つの検出済みクラスタを生成します。このうち、サービスクラスタのみがユーザーに表示されます。SNIP、HIP、および BE クラスタは、サービスクラスタの関連クラスタとして表示されます。HIP および BE クラスタは、*lb* 範囲に HIP と BE が存在する場合にのみ生成されます。

上記の例では、自動ポリシー検出により、サービスの SNIP と HIP を含む *lb* 範囲に SNIP クラスタと HIP クラスタが生成されます。BE は *lb* 範囲の外にあるため、自動ポリシー検出でバックエンドクラスタは生成されませんが、代わりに *db* の関連クラスタのリストに *be* 範囲が追加されます。

クラスタは次のように生成されます。

- サービス

  サービスクラスタには、サービスの VIP が含まれます。サービスクラスタのクエリは次のとおりです。

  ```
  user_orchestrator_system/cluster_id eq 1234 and
  user_orchestrator_system/namespace eq common and
  user_orchestrator_system/service_name eq db
  ```

- SNIP

  サービスの SNIP は、SNIP クラスタに含まれています。SNIP クラスタのクエリは次のとおりです。

  ```
  user_orchestrator_system/cluster_id eq 1234 and
  user_orchestrator_system/service_startpoint eq db
  ```

- HIP

サービスの HIP は、HIP クラスタに含まれています。HIP クラスタのクエリは次のとおりです。

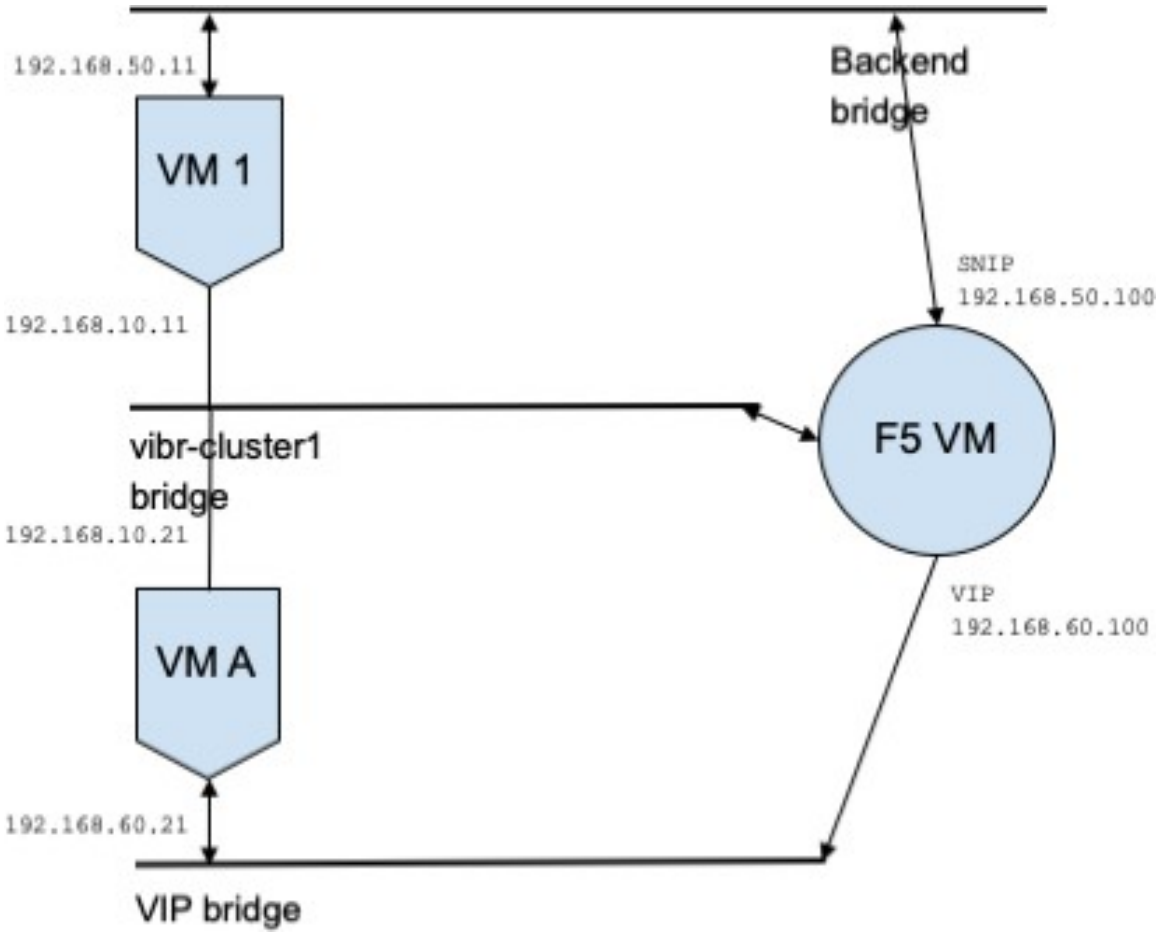
```
user_orchestrator_system/cluster_id eq 1234 and
user_orchestrator_system/service_healthcheck_startpoint eq db
```

- バックエンド

  1 つ以上の BE が *lb* 範囲の一部である場合、サービスのバックエンドクラスタが生成されます。これは上記の例には当てはまらないため、*lb* 範囲ではバックエンドクラスタが生成されません。

ポリシー

図 *356* : ポリシーの生成



VIP アドレス *192.168.60.100*、SNIP アドレス *192.168.50.100* を持つサービス *db* があるとします。また、IP アドレス *192.168.50.11* を持つバックエンド VM がポート 10000 でリッスンしています。クライアント VM *192.168.60.21* から *db* へのトラフィックのポリシーは次のようになります。

• クライアントから VIP へのポリシー

次のポリシーは、クライアント VM からサービス *db* へのアクセスを許可します。

```
{
"src": "<uuid of client scope>",
"dst": "<uuid of service cluster>",
"l4_params": [
{
"port": [
10000,
10000
],
"proto": 6,
}
]
}
```

• SNIP から BE へのポリシー。

SNIP から BE へのトラフィックを許可するポリシーは、設定に基づいて自動生成され、*db* の関連ポリシーとして表示されます。

```
{
"src": "<uuid of SNIP cluster>",
"dst": "<uuid of be scope>",
"l4_params": [
{
"port": [
10000,
10000
],
"proto": 6,
}
]
```

lb 範囲から *be* 範囲へのポリシーコネクタは、次のポリシーをプッシュします。

コンシューマ	プロバイダー	ポート	プロトコル	アクション
SNIP	be	10000	TCP	許可

これにより、BE ホスト 192.168.50.11 にファイアウォールルールが生成され、ポート 10000 の LB SNIP 192.168.50.100 からの着信トラフィックが許可されます。

• HIP から BE へのポリシー。

HIP から BE へのトラフィックを許可するポリシーは、設定に基づいて自動生成され、*db* の関連ポリシーとして表示されます。

```
{
"src": "<uuid of HIP cluster>",
"dst": "<uuid of be scope>",
"l4_params": [
{
"port": [
0,
0
],
"proto": ICMP,
}
```

```
]
}
```

lb 範囲から *be* 範囲へのポリシーコネクタは、次のポリシーをプッシュします。

コンシューマ	プロバイダー	ポート	プロトコル	アクション
HIP	be	0	ICMP	許可

これにより、BE ホスト *192.168.50.11* にファイアウォール ルールが生成され、LB HIP *192.168.50.2* からの着信 ICMP トラフィックが許可されます。

警告

- 同じロードバランサインスタンスからの複数のサービスが同じ名前を持つ場合、これらのサービスのいずれかに対して生成されるバックエンドルールには、それらすべてのバックエンドプールが含まれます。つまり、ルールは必要以上に寛容になります。

ポリシーパブリッシャ

ポリシーパブリッシャは Cisco Secure Workload の高度な機能であり、サードパーティベンダーは、ロードバランサやファイアウォールなどのネットワークアプライアンス向けに最適化された独自の適用アルゴリズムを実装できます。この機能は、定義済みのポリシーを Secure Workload クラスタ内にある Kafka インスタンスに公開し、お客様に Kafka クライアント証明書を提供することによって実現されます。これにより、サードパーティー ベンダー コードは Kafka からポリシーを取得でき、これらのポリシーを独自のネットワークアプライアンス構成に適宜変換できます。

このセクションの目的は、サードパーテベンダー、つまり以下におけるユーザーが Linux 上の Java でポリシーパブリッシャ機能を利用するために実行する必要がある手順を説明することです。

前提条件

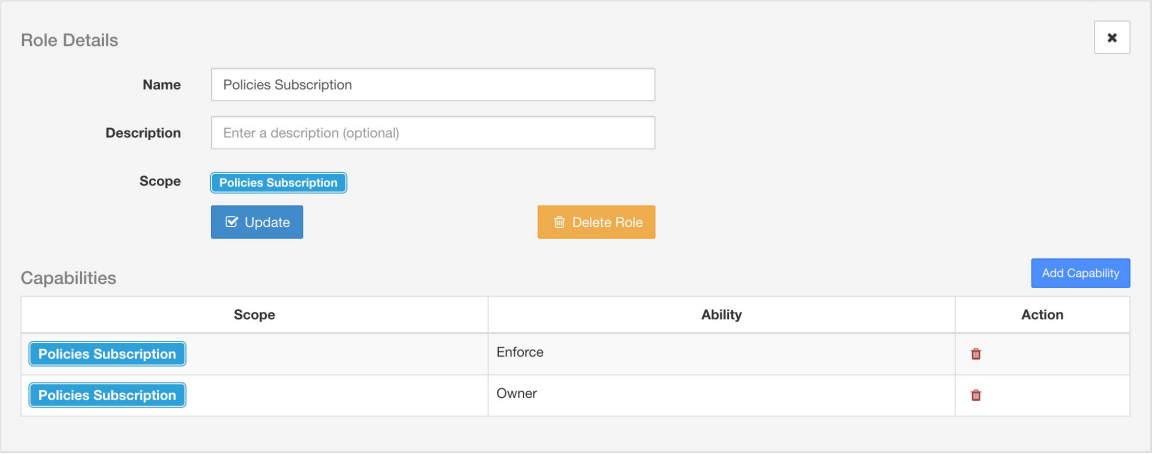
Ubuntu 16.04 などを搭載している Linux システムに次のソフトウェアパッケージがインストールされていること：

- Java 8 JDK
- Apache Kafka クライアント：kafka-clients-1.0.0.jar
- プロトコルバッファコア：protobuf-java-3.4.1.jar
- Apache Log4j：log4j-1.2.17.jar
- Java 用のシンプルなロギングファサード：slf4j-api-1.7.25.jar, slf4j-log4j12-1.7.25.jar
- Java 用のシンプルなコンプレッサ/デコンプレッサ：snappy-java-1.1.4.jar

Kafka クライアント証明書の取得

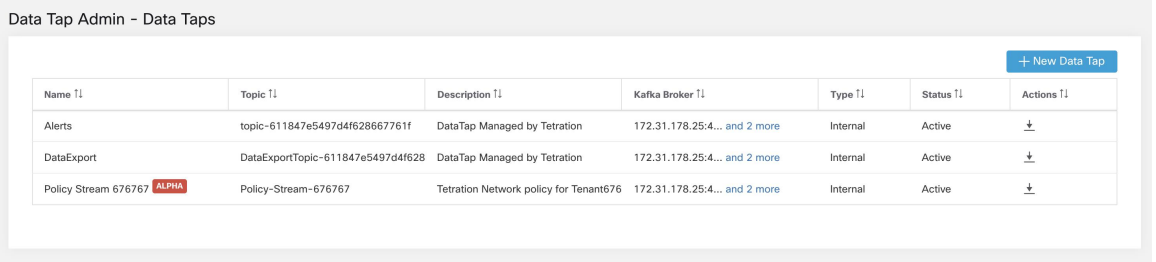
- 「所有者」のケーパビリティを持つユーザー ロールを作成し、選択したユーザーアカウントに割り当てます。

図 *357 : Kafka* からポリシーを受け取るためのユーザーロール設定



- 「ポリシーの適用」の説明に従って、ポリシーの適用を実行します。この最初のステップは、アクティブな範囲に関連付けられた Kafka トピックを作成するために必要です。
- **[管理（Manage）] > [データタップ管理者（Data Tap Admin）]**に移動します。
- [データタップ（Data Taps）] タブを選択し、[アクション（Actions）] 列のダウンロードボタンをクリックして、Kafka クライアント証明書をダウンロードします。ダウンロードダイアログで [Javaキーストア（Java Keystore）] フォーマットを選択してください。

図 *358 : [データタップ（Data Taps）] ビュー*



- ダウンロードされたクライアント証明書ファイルには、通常、*Policy-Stream-10-Policies-Subscription.jks.tar.gz* のような名前が付いています。ディレクトリを作成し、作成したディレクトリの下に以下のように解凍します。

```
    mkdir Policy-Stream-10-Policies-Subscription
    tar -C Policy-Stream-10-Policies-Subscription -zxf
Policy-Stream-10-Policies-Subscription.jks.tar.gz
```

Protobuf 定義ファイル

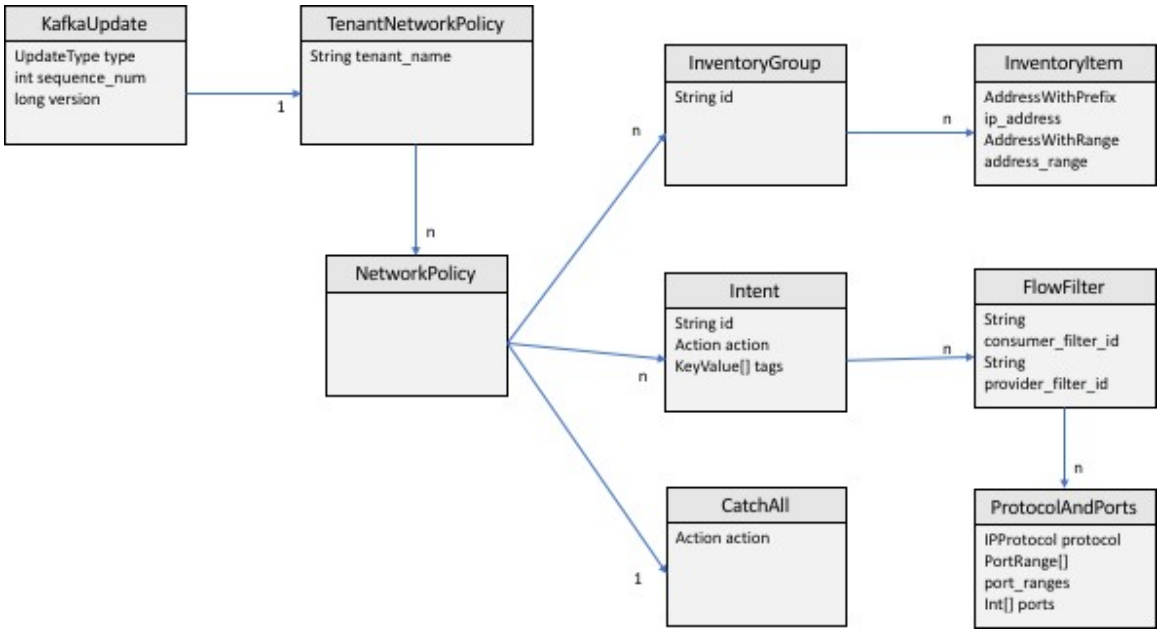
Secure Workload バックエンドによって Kafka に公開されるネットワークポリシーは、Google Protocol Buffers 形式でエンコードされます。Linux システムにダウンロードしてインストールする方法については、こちらのガイドを参照してください。

Secure Workload ネットワークポリシーの proto ファイルは、こちらからダウンロードできます。

Secure Workload ネットワークポリシーのデータモデル

下の図は、Kafka に接続されている Secure Workload エンティティの簡略化された UML ダイアグラムを示しています。

図 *359 : Secure Workload* ネットワークポリシーのデータモデル



protobuf でモデル化された Secure Workload ネットワークポリシーは、InventoryGroups のリスト、Intents のリスト、および CatchAll ポリシーで構成されます。各ポリシーには、1 つのルート範囲に属するすべての項目が含まれています。InventoryGroup には、単一のネットワークドレス、サブネット、またはアドレス範囲など、ネットワークアドレスを指定することによってサーバーやアプライアンスなどの Secure Workload エンティティを表す InventoryItems のリストが含まれます。Intent は、ネットワークフローが特定のコンシューマの InventoryGroup、プロバイダーの InventoryGroup、およびネットワークプロトコルとポートと一致するときに実行されるアクション（許可または拒否）を記述します。CatchAll は、Secure Workload 内部のルート範囲に対して定義されたキャッチオールアクションを表します。適用が有効になっているワークスペースがルート範囲に存在しない場合、デフォルトポリシーである ALLOW が生成されたポリシーに書き込まれます。

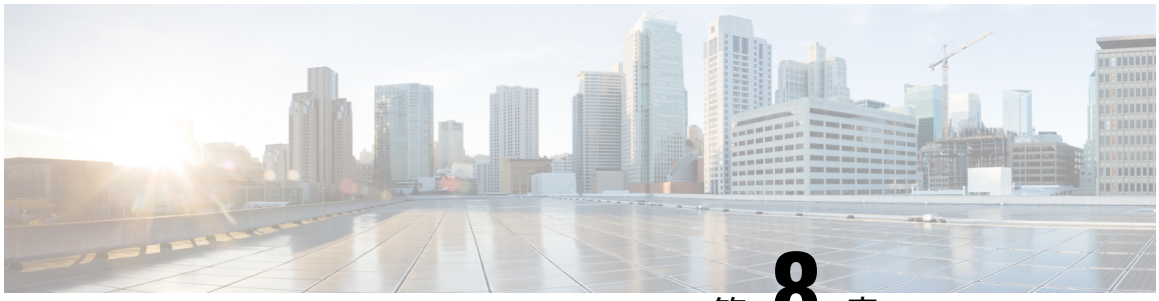
ユーザーまたはインベントリグループの変更によって適用がトリガーされると、Secure Workload バックエンドは、定義されたネットワークポリシーの完全なスナップショットを、KafkaUpdates として表される一連のメッセージとして Kafka に送信します。これらのメッセージを完全なスナップショットに再構築する方法と、エラー状態を処理する方法の詳細については、tetration_network_policy.proto ファイル内の KafkaUpdate のコメントを参照してください。

KafkaUpdate メッセージのサイズが 10MB を超える場合、Secure Workload バックエンドはこのメッセージをそれぞれのサイズが 10MB の複数のフラグメントに分割します。複数のフラグメントでは、最初のフラグメントのみに TenantNetworkPolicy の ScopeInfo フィールドがあります。ScopeInfo は、KafkaUpdate メッセージの残りのフラグメントで nil に設定されます。

Secure Workload ネットワーク ポリシー クライアントのリファレンス実装

リファレンス実装とデモクライアントをコンパイルして実行する方法については、Java でのこの tnp-enforcement-client を参照してください。

この実装により、Kafka のみを介して Secure Workload ポリシーストリームからネットワークポリシーを読み取るための共通コードが提供されます。実際のポリシーをネットワークデバイスにプログラムするベンダー固有のコードは、必要なインターフェイス PolicyEnforcementClient を実装することでプラグインできます。



第 8 章

フォレンジック

フォレンジック機能セットは、リアルタイムのフォレンジックイベントをキャプチャし、ユーザー定義のルールを適用することにより、起こり得るセキュリティインシデントの監視とアラートを可能にします。具体的には、次のことが可能になります。

- 関心のあるフォレンジックイベントを指定するルールの定義
- 一致するフォレンジックイベントに対するトリガーアクションの定義
- 特定のフォレンジック イベントの検索
- イベント生成プロセスとその完全な系統の可視化

警告 **フォレンジック**機能が有効になっている場合、センサーはセンサー構成に応じて追加のホストリソースを消費する場合があります。「ソフトウェアエージェントの設定」セクションを参照してください。

- 互換性 （698 ページ）
- フォレンジックシグナル （699 ページ）
- フォレンジック設定 （704 ページ）
- 法医学の可視化 （720 ページ）
- フォレンジックイベントに表示されるフィールド （723 ページ）
- フォレンジック分析：検索可能なフィールド （730 ページ）
- フォレンジック分析の検索用語 （730 ページ）
- フォレンジックアラート （737 ページ）
- フォレンジックスコア （740 ページ）
- PCR ベースのネットワーク異常検出 （742 ページ）
- プロセスハッシュの異常検出 （749 ページ）

互換性

フォレンジックシグナルは、AIXを除くすべてのプラットフォームの優れた可視性エージェントによって報告されます。詳細については、下記の「フォレンジックシグナル」のセクションを参照してください。

フォレンジック情報は、Linux カーネル API、監査と syslog、Windows カーネル API、Windows イベントなどを通じて提供されます。一般に、OS ベンダーはメジャーリリース内での互換性を保証します。ただし、OSベンダーが機能と修正をバックポートする可能性があるため、API はプラットフォームやマイナーリリース間でわずかに異なる可能性があります。その結果、一部のプラットフォームでは、一部のフォレンジック イベント タイプを使用できない場合があります。また、エージェントは起動時に無効になっている OS サービスの回復や有効化を試みません。

たとえば、Linux 監査フレームワークを使用するフォレンジックシグナルは数多くあります。フォレンジックが有効になっている場合、優れた可視性エージェントは、エージェントの起動後に Secure Workload の監査ルールをシステムに挿入します。ルールを挿入するには、システムにaugenrulesユーティリティがインストールされている必要があります。また、/etc/audit/rules.dディレクトリが必要です。これらの前提条件のいずれかが満たされていない場合、Secure Workloadの監査ルールは挿入されません。その結果、ファイルアクセスとrawソケット作成を含むフォレンジックシグナルは報告されません。

ユーザーが以前にフォレンジックを有効にしてから無効にした場合、センサーは Cisco Secure Workload によって挿入された監査ルールを削除します。Redhat 7.3 および CentOS 7.3 では、ルールの削除プロセスに影響を与える可能性のあるオペレーティングシステムのバグが確認されました。センサーが監査ルールを削除する過程は次のとおりです。 1. センサーは、*/etc/audit/rules.d/* の taau.rules を削除します。2. センサーは *$service auditd restart* を実行します。OS は、*/etc/audit/rules.d/* 内の *audit.rules* および **.rules* ファイルに基づいてルールセットを再生成します。次に、auditd はルールをシステムにロードします。

オペレーティング システムは、新しいルール セットを挿入する前にすべてのルールをクリアするために、/etc/audit/rules.d/audit.rules ファイルの先頭に -D を追加します。ただし、Redhat 7.3 および CentOS 7.3 マシンでは、*/etc/audit/rules.d/audit.rules* に *-D* がない場合があります。*/etc/audit/rules.d/audit.rules* ファイルが存在せず、*/usr/share/doc/audit-<version>/* のサブディレクトリにデフォルトのルールファイルも存在しない場合、OS が空のファイルを作成するため、このようなことが起こります。*/usr/share/doc/audit- 2.8.4/rules/10-base-config.rules* は、考えられるデフォルトのルールの格納場所の 1 つです。正確な OS の動作は、*$rpm -qf -scripts /etc/audit/rules.d* を実行して RPM 更新スクリプトから確認できます。

Linuxでは、一部のフォレンジックシグナルは64ビットシステムコールの監視に依存しています。32 ビット Linux システムコールは、現在のリリースではサポートされていません。

フォレンジックシグナル

ソフトウェアセンサーがフォレンジックイベントをキャプチャしてレポートするには、フォレンジック機能を有効にする必要があります。この機能は、[ソフトウェアエージェント構成（Software Agent Config）] で有効にできます。詳細については、「ソフトウェアエージェントの設定」を参照してください。

フォレンジック機能が有効になっている場合、センサーから次のフォレンジックイベントが報告されます。

信号	説明
特権昇格	Sudo で実行されるコマンドなどの特権昇格
ユーザーログオン	ユーザーログオンイベント
ユーザーログオン失敗	失敗したユーザーログオンの試行
シェルコード	シェルコードの試みに似た不審なシェル実行
File Access	パスワードファイルなどの非常に機密性の高いファイルへのアクセス
ユーザ アカウント	ユーザーアカウントの追加または削除
未確認コマンド	センサーが認識していない新しいコマンド。ユーザーはコマンド異常スコアを使用して、範囲に基づいて結果を調整できます。詳細については、「未確認コマンド」を参照してください。
未確認ライブラリ	以前にプロセスがロードされたことをセンサーが認識していない新しいライブラリ
raw ソケットの作成	raw ソケットを作成するプロセス（ポートノッキングなど）
バイナリ変更	既知のバイナリのハッシュ値または変更時刻の変更
ライブラリ変更	既知のライブラリのハッシュ値または変更時刻の変更
サイドチャネル	サイドチャネル攻撃の試み（Meltdown）
ユーザーログオンの追跡	ユーザーログオンイベント後に分岐または実行される子孫プロセス

信号	説明
移行手順	プロセスに基づくユーザーフォレンジック構成ルールに一致する移行手順イベントのレポートプロセス バイナリパス、コマンド文字列などの属性。
ネットワーク異常	ワークロードのネットワークトラフィックの異常。詳細については、「PCR ベースのネットワーク異常検出」を参照してください。

特権昇格

プロセスがその権限を低から高に変更すると、それは特権昇格と見なされます。Linux において、この変更は、プロセスのユーザーIDがゼロ以外からゼロに変更されたことを意味します。通常のユーザーのパスワードや、Sudo といった特殊目的バイナリの変更など、正当な場合があります。このイベントは現在、Windows では利用できません。Windows での特権昇格は、通常、プロセス自体の権限（整合性レベル）の変更ではなく、他のメカニズムを介して行われます。Windows での権限昇格は、未確認のコマンドや以下のバイナリ変更など、別タイプのフォレンジックイベントの対象となります。

ユーザーログオン

SSH、RDP、およびその他のタイプのログオンを含むユーザーログオンイベント。センサーは誰が、いつ、どのようにユーザーログインしたかを可能な場合は常にキャプチャします。たとえば、LinuxのSSHの場合、センサーはユーザー名、認証タイプ（パスワード、パブリック）、および送信元 IP を報告します。

ユーザーログオン失敗

前述のユーザーログオンイベントの場合と同様に、センサーは、同様の情報がある場合は常にログイン試行の失敗を報告します。

シェルコード

シェルコードイベントの解釈は、Linux と Windows で異なります。Linux では、センサーはログインセッションまたは端末なしで対話型シェルとして実行されているプロセスを識別します。（ログインセッションの外で対話型シェルを実行する特別な理由はありません。）このリリースでは、システムですでに利用可能なシェルを利用して攻撃されることを想定しているため、シェルコードイベントの検出は制限されています。攻撃によって新しいバイナリがアップロードされた場合、センサーはこれらのバイナリに、未知のコマンドまたはバイナリの変更（既存のバイナリを置き換える場合）のフラグを付けます。Windows では、PowerShell DLL に

リンクされているすべてのプロセスがシェルコードとしてラベル付けされます。ユーザーは、正当なケースを除外するルールを作成できます。

File Access

ファイルアクセスイベントは、パスワードファイルなどの非常に機密性の高いファイルへのアクセスを報告します。このリリースでは、ユーザーは監視対象ファイルのリストを変更できません。Linux では、センサーは /etc/passwd への書き込みアクセスを監視します。センサーは、/etc/shadow への読み取りおよび書き込みアクセスも監視します。このリリースでは、Windows はこのイベントをトリガーしません。

ユーザーアカウント

ユーザーアカウントイベントは、情報が利用可能な場合はいつでも、ローカルユーザーアカウントの作成をレポートします。

未確認コマンド

未確認コマンドイベントは、センサーがこれまで認識していなかったコマンドを報告します。未確認コマンドは、親プロセスから子プロセスへの目に見えない移行/エッジとして定義されます。たとえば、Web サーバー（httpd）が abc.sh という CGI スクリプトを実行していると仮定すると、センサーはそれを初めて検出したときに、abc.sh を未確認コマンドとして報告します。Web サーバーによるそれ以降の abc.sh の実行では、センサーが以前にそれを検出して報告しているため、フォレンジックイベントは発生しません。サービスまたはプロセスがバイナリを実行しない場合、そのサービス/プロセスからの未確認コマンドイベントは、侵害である可能性を示します。センサーは再起動後もステートレスであるため、センサーの再起動後、以前確認されたコマンドが再度報告されることに注意してください。

SaaS クラスタ 3.4 以降の場合、それぞれの未確認コマンドイベントは、0.0 から 1.0 の範囲のコマンド異常スコアに関連付けられています。スコアが低いほど、移行の異常性が高くなります。コマンド移行、つまりタプル（親コマンドライン、コマンドライン）は、以下の同じタプルを持つイベント間の異常な移行についてクロスチェックされます。

- センサーが属する最も狭い範囲。たとえば、次の範囲系統 `Root Scope -> A -> B -> C and Root Scope -> D -> E` に属するワークロード W において、未確認コマンドイベントが観察されます。その後、コマンドは範囲 C と E のすべてのワークロード間でクロスチェックされます（C と E はオーバーラップしている場合もしていない場合もあることに注意してください）。イベントの異常スコアは、これら 2 つの範囲に関するイベントの異常スコアの最大値です。
- 実行中のプロセスの実行パス。
- 親プロセスの実行パス。
- 実行中のプロセスのバイナリハッシュ。

スコア 1.0 は、同じタプル（最も狭い範囲、実行パス、親実行パス、バイナリハッシュ）を持つ、同じコマンド移行が表示されたことを意味します。スコア 0.0 は、実行パス、親実行パス、および実行中のプロセスのバイナリハッシュを使用したコマンド移行が、同じ範囲内のどのホストでも観察されていないことを意味します。異常スコアを使用して、同様の未確認コマンドアラートが同じ範囲内で発生するのを抑制し、誤検知を減らすことができます。このスコアの使用例については、「デフォルトの Secure Workload ルール」を参照してください。

異常スコアは、3.4 の SaaS クラスタでのみ使用できることに注意してください。

未確認ライブラリ

未確認ライブラリイベントは、センサーが以前にロードされたプロセスを認識していないライブラリを報告します。未確認ライブラリは、バイナリ実行パスとライブラリパスの未確認ペアとして定義されます。たとえば、アプリケーションは通常、比較的安定したライブラリのリストをロードします。マシンにアクセスできる攻撃者は、アプリケーションを再起動し、悪意のあるライブラリで LD_PRELOAD を使用する可能性があります。センサーは、このアプリケーションのバイナリ実行パスに新しくロードされた悪意のあるライブラリを初めて検出すると、未確認ライブラリイベントを報告します。センサーが以前に検出して報告しているため、悪意のあるライブラリの後続ロードによってフォレンジックイベントが発生することはありません。正当なケースには、アップグレード後にアプリケーションが新しいライブラリをロードする場合や、アプリケーションが新しいライブラリを動的にロードする場合があります。再起動後、センサーが以前に確認されたライブラリを再度報告する場合があることに注意してください。

これは実験的な機能であり、将来のリリースで変更される可能性があります。

raw ソケットの作成

raw ソケット作成イベントは、このリリースの Linux でのみサポートされています。raw ソケットは通常、トラフィックのスヌーピング、インジェクション、スプーフィングに使用されます。raw ソケットの正当な用途には、 診断ツール（tcpdump）や、 特別な IP パケット（ping、arp）を生成する場合などがあります。ターゲットマシンや被害を受けたマシンによるロギングを回避するステルススキャン、マルウェアポートノッキングなど、悪意を持って使われる場合もあります。また、Secure Workload センサーもフロー関連情報を収集するための raw ソケットを作成します（一貫性を確保するため、センサーは自身のフロー情報収集によってトリガーされたイベントを抑制しません）。

バイナリ変更

バイナリ変更イベントは、実行中のプロセスのファイルコンテンツとバイナリの属性に対する変更を報告します。センサーは、実行中のすべてのプロセスのファイル属性を記録します。プロセスが同じパスでバイナリを実行しているが、ファイル属性（ctime、mtime、サイズ、またはハッシュ）が異なる場合、センサーはプロセスにバイナリ変更のフラグを付けます。正当なケースには、アプリケーションのアップグレードが含まれます。

ライブラリ変更

ライブラリ変更イベントは、実行中のプロセスのファイルコンテンツとライブラリの属性に対する変更を報告します。センサーには、ロードされたライブラリのファイル属性が記録されます。プロセスにより同じパスでライブラリがロードされるが、ファイル属性（ctime、mtime、サイズ、またはハッシュ）が異なる場合、センサーによりプロセスにライブラリ変更のフラグが付けられます。正当なケースには、ライブラリのアップグレードが含まれます。

これは実験的な機能であり、将来のリリースで変更される可能性があります。

サイドチャネル

サイドチャネルイベントは、サイドチャネルの脆弱性をエクスプロイトする実行中のソフトウェアを報告します。このリリースは、選択された Linux プラットフォームでの 1 つのサイドチャネル検出機能を提供します。Meltdown。サポートされているマシン構成については、以下の詳細を参照してください。これらは高度なセキュリティ機能であるため、デフォルトで無効になっています。サイドチャネルレポートが有効になっている場合、CPU 使用率の増加が予想されます。UI で設定された CPU クォータは引き続き適用されます。センサーのフォレンジック コレクション サブプロセスにより、CPU 使用率が高すぎると判断された場合、センサーがシャットダウンし、親センサープロセスが少し遅れて再起動します。古いカーネルやサポートされていないカーネルでこの機能を有効にすると、システムが不安定になる可能性があります。同様の非本番環境でテストすることを強く推奨します。

この機能は、UI の [エージェント構成（Agent Config）] ページからオン/オフの切り替えができ、各エージェント構成プロファイルでオン/オフを切り替えられます。

Meltdown は、CPU の投機的実行とキャッシュ機能を悪用するサイドチャネル攻撃です（https://meltdownattack.com/）。攻撃者は非特権ドメインから特権ドメインデータを読み取ることができます。たとえば、リング 0 特権なしでユーザー空間アプリケーションからカーネルメモリを読み取ることができます。Meltdown 検出は現在、CentOS 7 および Ubuntu 16.04 をサポートしています。

ユーザーログオンの追跡

ユーザーログオンイベントの追跡では、ユーザー ログオン イベント プロセス（SSH、RDP など）の後に実行される子孫プロセス (最大 4 レベル) が報告されます。このユーザーログオンイベントの追跡で報告されるプロセスは監査目的であり、セキュリティイベントは必要ありません。

移行手順

プロセス後イベントは、バイナリパス、コマンド文字列などのプロセス属性に基づいて、ユーザーのフォレンジック構成ルールに一致するプロセスをレポートします。このプロセス後イベントの下でレポートされるプロセスは、監査目的のためのものであり、必ずしもセキュリティイベントを持つとは限りません。

例 1：cmd.exe または powershell.exe によって実行されたプロセスを報告する

Event Type = Follow Process AND (Process Info - Exec Path contains cmd.exe OR Process Info - Exec Path contains powershell.exe)

例 2：winword.exe、excel.exe、または powerpnt.exe によって作成されたプロセスを報告する

Event Type = Follow Process with_ancestor (Process Info - Exec Path contains winword.exe OR Process Info - Exec Path contains excel.exe OR Process Info - Exec Path contains powerpnt.exe)

注：プロセス後イベントは、次のプロセスシグナルのいずれかによって追跡できます。

- Process Info - Exec Path
- Process Info - Command String
- Process Info - Username
- プロセス後 - 親 Exec パス
- プロセス後 - 親コマンド文字列
- プロセス後 - 親ユーザー名

フォレンジック設定

フォレンジック設定では、インテントベースの設定が使用されます。インテントは、フォレンジックプロファイルをインベントリフィルタに適用する方法を指定します。フォレンジックプロファイルは、複数のフォレンジックルールで構成されます。インテント内のプロファイルは、上から順に適用されることに注意してください。

フォレンジックルール



（注） ルート範囲あたりのルールの最大数は 100 です。

フォレンジックルールの追加

このセクションでは、新しいフォレンジックルールを追加する方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

手順

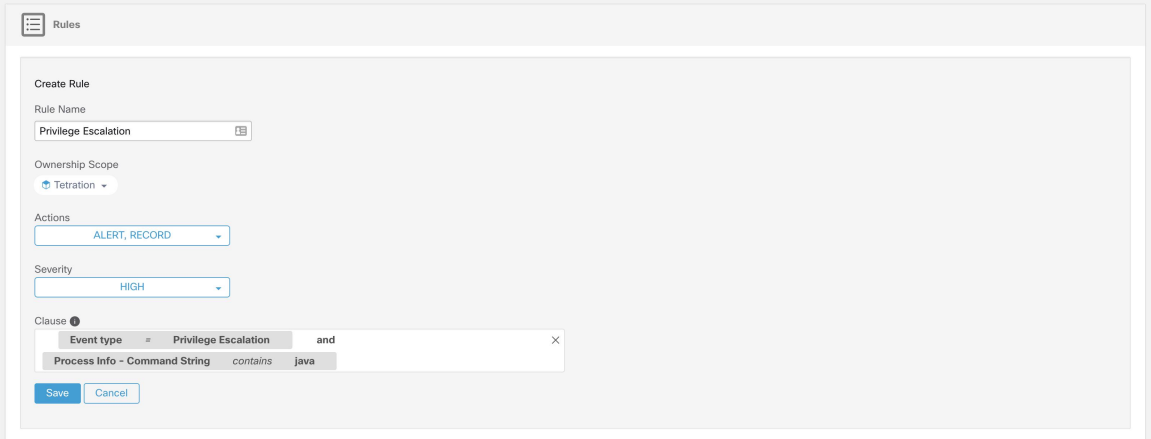
ステップ 1 左側のナビゲーションバーで、**[防御（Defend）]** > **[フォレンジックルール（Forensic Rules）]** をクリックします。

ステップ 2 [ルールの作成（Create Rule）] をクリックします。

ステップ 3 以下のフィールドに適切な値を入力します。

フィールド	説明
ルール名（Rule Name）	ルールの名前を入力します。名前は空白にできません。
[所有範囲（Ownership scope）]	このルールの所有範囲を入力します。
アクション（Actions）	このルールがトリガーされたときのアクションを選択します。[記録（Record）] は、一致するセキュリティイベントをさらに分析するために保持することを意味します。[アラート（Alert）] アクションは、一致するセキュリティイベントを Secure Workload アラートシステムに公開することを意味します。
重大度	このルールのシビラティ（重大度）レベルを選択します（[低（LOW）]、[中（MEDIUM）]、[高（HIGH）]、[重大（CRITICAL）]、[ただちに対応が必要（REQUIRES IMMEDIATE ACTION）]）。
[句（Clause）]	ルール句を入力します。句には、プロセスフォレンジック イベントまたはワークロードイベントのいずれかからの、セキュリティ イベント シグナルが含まれている必要があります。プロセスシグナルとワークロードシグナルの両方が含まれている句は無効です。

図 *360* : ルールの作成



ステップ 4 [保存（Save）] をクリックします。

基本的なフォレンジックルールの構成

フォレンジックルールには、フォレンジックイベントタイプを**1つだけ**指定する必要があります（例：**Event Type == Unseen Command**）。次のオプション句では、そのイベントの属性を使用する必要があります（例：**Unseen Command - Parent Uptime**）。

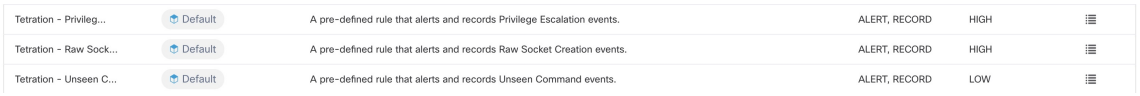
以下は、**Unseen Command** イベントタイプを使用した1つの例です。その他の例については、下記のデフォルトルールと MITER ルールをご覧ください。

EventType = Unseen Command and Unseen Command - Parent Uptime (microseconds) >= 60000000.

デフォルトの Secure Workload ルール

ユーザーが自分の環境に合ったルールを構築する際の手助けとなるように、デフォルトの Secure Workload ルールが用意されています。これらのルールは、フォレンジック設定ページに表示され、編集できません。ルールはすべてのルート範囲で使用できます。

図 *361* : デフォルトルール



このリリースには、4 つの Secure Workload フォレンジックルールが組み込まれています。

1. **ルール名** Secure Workload - 特権昇格

   Clause EventType = Privlege Escalation and (ProcessInfo - ExecPath *doesn't contain* **sudo and ProcessInfo - ExecPath** *doesn't contain* **ping and Privilege Escalation Is≠ Type - Suid Binary)**

   説明：このルールは、setuid バイナリによって生成されない特権昇格イベントを報告します。setuid バイナリを確実に除外するためには、「ProcessInfo - ExecPath」に基づいて

sudo と **ping** も除外します。Secure Workload ユーザーは、独自のルールを定義することで、他の setuid バイナリを除外することもできます

2. **ルール名** Tetration - 未確認のコマンド

   Clause EventType = Unseen Command and Unseen Command - Parent Uptime (microseconds) >= 60000000 or ProcessInfo - ExecPath *contains* **/bash or ProcessInfo - ExecPath** *contains* **/sh or ProcessInfo - ExecPath** *contains* **/ksh or Parent - ExecPath** *contains* **httpd or Parent - ExecPath** *contains* **apache or Parent - ExecPath** *contains* **nginx or Parent - ExecPath** *contains* **haproxy**

   説明：このルールは、次の条件のいずれかに一致する未確認のコマンドイベントを報告します。

   1. プロセスの親が **60,000,000** マイクロ秒を超えて稼働している
   2. プロセス ExecPath には、何らかのタイプのシェル（**/bash**、**/sh**、**/ksh** など）が含まれている。
   3. プロセスの親 ExecPath には、何らかのタイプのサーバーアプリケーション（**httpd**、**apache**、**nginx**、**haproxy** など）が含まれている。

3. **ルール名** Tetration - raw ソケット

   Clause EventType = Raw Socket Creation and (Raw Socket - ExecPath *doesn't contain* **ping and Raw Socket - ExecPath** *doesn't contain* **iptables and Raw Socket - ExecPath** *doesn't contain* **xtables-multi**)

   説明：このルールは、**ping** や **iptables** によって生成されない raw ソケット作成イベントを報告します。Secure Workload ユーザーは、独自のルールを定義して、他のバイナリを除外することもできます。

4. **ルール名** Tetration - 未確認コマンドによるネットワーク異常

   Clause EventType = Network Anomaly and Network Anomaly - Unseen Command Count > 3 and Network Anomaly - Non-seasonal Deviation > 0

   説明：このルールは、次の条件に一致するネットワーク異常イベントを報告します。

   1. 15 分以内に同じワークロードで 3 つ以上の未確認コマンドイベントがある。
   2. ルールの属性が 0 より大きい（6.0 はすべてのネットワーク異常イベントについて報告される最小偏差であるため、6.0 以上であることも意味します）。

5. **ルール名** Tetration - 異常な未確認コマンド

   Clause EventType = Unseen Command and Unseen Command - Anomaly - Score < 0.6

   説明：このルールは、異常スコアが 0.6 未満の未確認のコマンドイベントを報告します。これは、コマンドが以前に観察されたコマンドと類似していないように見える非常に異常なイベントのみが報告されることを意味します。しきい値 0.6 は、類似コマンドが異なるしきい値でどのように動作するかに関する Cisco Secure Workload での実験に基づいて決定されます。スコアの詳細な説明については、「未確認コマンド」を参照してください。

6. **ルール名** Tetration - SMSS の異常な親

   Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **smss.exe and (Follow Process - ParentExecPath** *doesn't contain* **smss.exe and Follow Process - ParentExecPath** *doesn't contain* **System**)

   説明：これは Windows 独自のルールです。このルールは、smss.exe の別のインスタンスまたはシステムプロセスとは異なる親が smss.exe にある場合に警告します。

7. **ルール名** Tetration - wininit の異常な親

   Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **wininit.exe and Follow Process - ParentExecPath** *doesn't contain* **smss.exe**

   説明：これは Windows 独自のルールです。このルールは、wininit.exe に smss.exe とは異なる親がある場合に警告します。

8. **ルール名** Tetration - RuntimeBroker の異常な親

   Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **RuntimeBroker.exe and Follow Process - ParentExecPath** *doesn't contain* **svchost.exe**

   説明：これは Windows 独自のルールです。このルールは、RuntimeBroker.exe に svchost.exe とは異なる親がある場合に警告します。

9. **ルール名** Tetration - サービスの異常な親

   Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **services.exe and Follow Process - ParentExecPath** *doesn't contain* **wininit.exe**

   説明：これは Windows 独自のルールです。このルールは、services.exe に wininit.exe とは異なる親がある場合に警告します。

10. **ルール名** Tetration - lsaio の異常な親

    Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **lsaio.exe and Follow Process - ParentExecPath** *doesn't contain* **wininit.exe**

    説明：これは Windows 独自のルールです。このルールは、lsaio.exe に wininit.exe とは異なる親がある場合に警告します。

11. **ルール名** Tetration - lsass の異常な子

    Clause (EventType = Follow Process and ProcessInfo - ExecPath *doesn't contain* **efsui.exe and ProcessInfo - ExecPath** *doesn't contain* **werfault.exe) with ancestor Process Info - ExecPath** *contains* **lsass.exe**

    説明：これは Windows 独自のルールです。このルールは、lsass.exe に efsui.exe または werfault.exe 以外の子孫がある場合に警告します。

デフォルトの MITRE ATT&CK ルール

デフォルトの MITRE ATT&CK ルールは、MITRE ATT&CK フレームワーク（https://attack.mitre.org/）からのアラート手法に対して指定されています。安全を脅かす行動に関連した 24 のルールがあり、それらのほとんどは特定の MITER 手法にマッピングされています。ルールの完全なリストを以下に示します。

1. **名前** 疑わしい MS Office の動作

   Clause (Event type = Follow Process and (Process Info - Exec Path doesn't contain **Windowssplwow64.exe) and (Process Info - Exec Path** *doesn't contain* **chrome.exe) and (Process Info - Exec Path** *doesn't contain* **msip.executionhost.exe) and (Process Info - Exec Path** *doesn't contain* **msip.executionhost32.exe) and (Process Info - Exec Path** *doesn't contain* **msosync.exe) and (Process Info - Exec Path** *doesn't contain* **ofccccaupdate.exe) with ancestor (Process Info - Exec Path** *contains* **winword.exe or Process Info -Exec Path** *contains* **excel.exe or Process Info -Exec Path contains powerpnt.exe)**

   説明 このルールは、Microsoft Office プロセス（WIN-WORD.exe/EXCEL.exe/POWERPNT.exe）が子プロセスを作成した場合にアラートを出し、記録します。シスコの調査に基づき、誤検知の量を減らすため、これらの MS Office バイナリによって作成されることが知られているいくつかの一般的な子プロセスを許可しています。

2. **名前** T1015 - アクセシビリティ機能 1

   Clause Event type = Follow Process (Process Info - Exec Path *contains* **cmd.exe or Process Info -Exec Path** *contains* **powershell.exe or Process Info - Exec Path** *contains* **cscript.exe or Process Info - Exec Path** *contains* **wscript.exe) and (Follow Process - Parent Exec Path** *contains* **winlogon.exe or Follow Process - Parent Exec Path** *contains* **atbroker.exe or Follow Process - Parent Exec Path** *contains* **utilman.exe**)

   説明 このルールは、アクセシビリティ機能のバイナリ（スクリーンキーボード、拡大鏡、固定キーなど）のいずれかが悪用され、cmd/powershell/cscript/wscript を開くように誘導された場合にアラートを出し、記録します。アクセシビリティバイナリの呼び出しは、どこから呼び出されるか（ログオン画面から、またはユーザーのログイン後）に応じて、winlogon、atbroker、または utilman のいずれかのプロセスによって制御されます。このルールは、アクセシビリティプロセス（winlogon.exe、utilman.exe、および atbroker.exe）の疑わしい子プロセス（cmd.exe、powershell.exe、cscript.exe、wscript.exe）をキャプチャします。これを [T1015 -アクセシビリティ機能2（T1015 - Accessibility features 2）] で使用して、4 つの疑わしい子プロセスの追加の子プロセスも捕捉します。**

3. **名前** T1015 - アクセシビリティ機能 2

   Clause Event type = Follow Process with ancestor ((Process Info - Exec Path *contains* **cmd.exe or Process Info - Exec Path** *contains* **powershell.exe or Process Info - Exec Path** *contains* **cscript.exe or Process Info - Exec Path** *contains* **wscript.exe) and (Follow Process - Parent Exec Path** *contains* **winlogon.exe or Follow Process - Parent Exec Path** *contains* **atbroker.exe or Follow Process - Parent Exec Path** *contains* **utilman.exe))**

   説明 このルールは、アクセシビリティ機能のバイナリ（スクリーンキーボード、拡大鏡、固定キーなど）のいずれかが悪用され、cmd.exe/powershell.exe/cscript.exe/wscript.exe を開くように誘導された場合にアラートを出し、記録します。アクセシビリティバイナリの呼び出しは、どこから呼び出されるか（ログオン画面から、またはユーザーのログイン後）に応じて、winlogon、atbroker、または utilman のいずれかのプロセスによって制御されます。このルールは、これらのプロセス（winlogon、utilman、および atbroker）の疑わしい子プロセスの子プロセスをキャプチャします。アクセシビリティバイナリの疑わしい子プロセスについてアラートを出す [T1015 -アクセシビリティ機能1（T1015 - Accessibility features 1）] でこれを使用する必要があります。

4. **名前** T1085 - rundll32

   Clause (Event type = Follow Process and Process Info Exec Path *doesnt contain* **msiexec.exe and Process Info Exec Path** *doesnt contain* **WindowsSystem32SystemPropertiesRemote.exe with ancestor (Process Info - Exec Path** *contains* **rundll32.exe and Follow Process - Parent Exec Path** *doesnt contain* **msiexec.exe and not (Process Info -command string** *contains* **Windowssystem32shell32.dll or (Process Info -command string** *contains* **Windowssyswow64shell32.dll or (Process Info -command string** *contains* **WindowsSystem32migrationWinInetPlugin.dll))**

   説明 このルールは、rundll32.exeが子プロセスを作成した場合にアラートを出し、記録します。このバイナリは、任意のバイナリ/dll を実行するために呼び出されたり、悪意のあるコントロールパネル項目をインストールするために control.exe によって使用されたりする可能性があります。ただし、msiexec.exe が rundll32.exe の親または子である場合は許可しています。また、よく知られた dll を利用するいくつかの一般的な rundll32 コマンドも許可しています。

5. **名前** T1118 - InstallUtil

   Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **installutil.exe**

   説明 このルールは、InstallUtil.exe が子プロセスを作成した場合にアラートを出し、記録します。

6. **名前** T1121 - Regsvcs/Regasm

   Clause Event type = Follow Process and (Process Info - Exec path *doesn't contain* **fondue.exe or Process Info - Exec path** *doesnt contain* **regasm.exe or Process Info - Exec path** *doesnt contain* **regsvr32.exe with ancestor (Process Info - Exec Path** *contains* **regasm.exe or Process Info - Exec Path** *contains* **regsvcs.exe)**

   説明 このルールは、regsvcs.exe または regasm.exe が子プロセスを作成した場合にアラートを出し、記録します。ただし、誤検知の数を減らすため、fondue.exe/regasm.exe/regsvr32.exe が regasm.exe または regsvcs.exe によって生成される場合は許可しています。

7. **名前** T1127 - 信頼できる開発者ユーティリティ - msbuild.exe

   Clause (Event type = Unseen Command with ancestor Process Info - Exec Path *contains* **MSBuild.exe) and (Process Info - Exec Path** *doesn't contain* **Tracker.exe) and (Process Info -Exec Path** *doesn't contain* **csc.exe) and (Process Info - Exec Path** *doesn't contain* **Microsoft Visual Studio) and (Process Info - Exec Path** *doesn't contain* **al.exe) and (Process Info - Exec Path** *doesn't contain* **lc.exe) and (Process Info - Exec Path** *doesn't contain* **dotnet.exe) and (Process Info - Exec Path** *doesn't contain* **cvtres.exe) and (Process Info - Exec Path** *doesn't contain* **conhost.exe) and not (Event type = Unseen Command with ancestor (Process Info - Exec Path** *contains* **Tracker.exe or Process Info - Exec Path** *contains* csc.exe or Process Info - Exec Path *contains* **Microsoft Visual Studio or Process Info - Exec Path** *contains* **al.exe or Process Info - Exec Path** *contains* **lc.exe or Process Info - Exec Path** *contains* **dotnet.exe or Process Info - Exec Path** *contains* **cvtres.exe))**

   説明 このルールは、msbuild.exeが、通常作成する子プロセスの許可リストに属していない子プロセスを作成した場合にアラートを出し、記録します。Follow Processはプロセスサブツリーの許可をまだサポートしていないため、このルールは現在、Follow Processで

はなく Unseen Command に基づいています。現在のルールでは、次のプロセスとその子が許可されます。tracker.exe、csc.exe、「Microsoft Visual Studio」パスからのプロセス、al.exe、lc.exe、dotnet.exe、および cvtres.exe。このルールでは conhost.exe も許可されます。これらのプロセスは、MSBuild.exe の通常の使用中に表示されます（たとえば、Visual Studio を使用したプロジェクトのコンパイルなど）。MSBuild.exe の上記以外の子プロセスすべて（通常の動作ではない）に関してアラートが生成されます。

8. **名前** T1127 - 信頼できる開発者ユーティリティ - rcsi.exe

   Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **rcsi.exe**

   説明 このルールは、rcsi.exe が子プロセスを作成した場合にアラートを出し、記録します。

9. **名前** T1127 - 信頼できる開発者ユーティリティ - tracker.exe

   Clause (Event type = Unseen Command with_ancestor Process Info - Exec Path *contains* **tracker.exe) and not (Event type = Unseen Command with_ancestor Process Info - Exec Path** *contains* **MSBuild.exe)**

   説明 このルールは、tracker.exe が子プロセスを作成した状況で、tracker 自体が MSBuild.exe の子ではない場合にアラートを出し、記録します。したがって、Visual Studio を使用した tracker の正当な呼び出しは承認されますが、他の呼び出しにはアラートが出されます。tracker.exe ルールおよび以前の MSBuild.exe ルールの限界の 1 つに、攻撃者が MSBuild 手法を使用して tracker を作成し、次いで tracker に悪意のある子を作成させた場合、MSBuild を先祖とする tracker は正当であるため、どちらのルールによってもアラートが出されない点があります。

10. **名前** T1128 - Netsh ヘルパー Dll

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **netsh.exe**

    説明 このルールは、netsh.exe が子プロセスを作成した場合にアラートを出し、記録します。

11. **名前** T1136 - アカウントの作成

    Clause Event type = User Account

    説明 このルールは、新しいユーザーが作成された場合にアラートを出し、記録します。

12. **名前** T1138 - アプリケーションシミング

    Clause Event type = Follow Process Process Info - Exec Path *contains* **sdbinst.exe**

    説明 このルールは、sdbinst.exe が呼び出された場合にアラートを出し、記録します。

13. **名前** T1180 - スクリーンセーバー

    Clause Event type = Follow Process AND with ancestor Process Info - Exec Path *contains* **.scr**

    説明 このルールは、exec パスで「.scr」を使用してプロセスが作成された場合にアラートを出し、記録します。

14. **名前** T1191 - CMSTP

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **cmstp.exe**

説明 このルールは、cmstp.exeが子プロセスを作成した場合にアラートを出し、記録します。

15. **名前** T1202 - 間接コマンド実行 - forfiles.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **forfiles.exe**

    説明 このルールは、forfiles.exe が子プロセスを作成した場合にアラートを出し、記録します。

16. **名前** T1202 - 間接コマンド実行 - pcalua.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **pcalua.exe**

    説明 このルールは、pcalua.exe が子プロセスを作成した場合にアラートを出し、記録します。

17. **名前** T1216 - 署名付きスクリプトプロキシ実行 - pubprn.vbs

    Clause Event type = Follow Process with ancestor ((Process Info - Exec Path *contains* **cscript.exe or Process Info - Exec Path** *contains* **wscript.exe) and Process Info - Command String** *contains* **.vbs and Process Info - Command String** *contains* **script)**

    説明 このルールは、パラメータ「script」を含む wscript.exe または cscript.exe を使用して vbs スクリプトが実行され、新しいプロセスが作成された場合にアラートを出し、記録します。攻撃者はこの手法を使用して、コードを実行する悪意のある sct ファイルを指すスクリプトパラメータを含む pubprn.vbs を実行する可能性があります。

18. **名前** T1218 - 署名付きバイナリプロキシ実行 - msiexec.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **msiexec.exe**

    説明 このルールは、msiexec.exeが子プロセスを作成した場合にアラートを出し、記録します。

19. **名前** T1218 - 署名付きバイナリプロキシ実行 - odbcconf.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **odbcconf.exe**

    説明 このルールは、odbcconf.exe が子プロセスを作成した場合にアラートを出し、記録します。

20. **名前** T1218 - 署名付きバイナリプロキシ実行 - Register-CimProvider

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **Register-CimProvider.exe**

    説明 このルールは、Register-CimProvider.exe が子プロセスを作成した場合にアラートを出し、記録します。

21. **名前** T1220 - XSL スクリプト処理 - msxsl.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **msxsl.exe**

    説明 このルールは、msxsl.exeが子プロセスを作成した場合にアラートを出し、記録します。

22. **名前** T1220 - XSL スクリプト処理 - wmic

    Clause Event type = Follow Process and (Process Info - Exec Path *contains* **wmic.exe and Process Info - Command String** *contains* **.xsl)**

    説明 このルールは、xsl スクリプトが wmic によって使用されている場合にアラートを出し、記録します。これは任意のバイナリを起動するために使用できます。

23. **名前** T1223 - コンパイル済み HTML ファイル

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **hh.exe**

    説明 このルールは、hh.exe が子プロセスを作成した場合にアラートを出し、記録します。

24. **名前** T1003 - ログイン情報ダンピング - Lsass

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **procdump.exe and Process Info - Command String** *contains* **lsass**

    説明 このルールは、lsass プロセスのメモリをダンプするために procdump.exe が使用されている場合にアラートを出し、記録します。

25. **名前** T1140 - ファイルまたは情報の難読化解除/復号化

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **certutil.exe and (Process Info - Command String** *matches* **.*encode\s.* or Process Info - Command String** *matches* **.*decode\s.***

    説明 このルールは、ファイルのエンコードまたはデコードに certutil.exe が使用されている場合にアラートを出し、記録します。この手法は、攻撃者が攻撃対象のマシンでエンコードされたペイロードをデコードするためによく使用されます。

26. **名前** T1076 - リモート デスクトップ プロトコル

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **tscon.exe**

    説明 このルールは、tscon.exe が実行された場合にアラートを出し、記録します。攻撃者は、tscon.exe を使用して既存の RDP セッションをハイジャックできます。

27. **名前** T1197 - BITS ジョブ - Powershell

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **powershell.exe and Process Info - Command String** *contains* **Start-BitsTransfer**

    説明 このルールは、powershell.exe を使用して cmdlet Start-BitsTransfer を実行し、ファイルをコピー/移動した場合にアラートを出し、記録します。

28. **名前** T1170 - MSHTA

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **mshta.exe**

    説明 このルールは、子プロセスを生成する悪意のある HTA スクリプトを実行するために mshta.exe が使用されている場合にアラートを出し、記録します。

29. **名前** T1158 - 非表示ファイルおよびディレクトリ

Clause Event type = Follow Process and (Process Info - Exec Path *contains* **attrib.exe and Process Info - Command String** *contains* **+h)**

説明 このルールは、attrib.exe を使用してファイル/ディレクトリを非表示に設定した場合にアラートを出し、記録します。

30. **名前** T1114 - 電子メール収集

    Clause Event type = Follow Process (Process Info - Command String *matches* **.*.(ost|pst)(\s|"|').* or Process Info - Command String** *matches* **.*.(ost|pst)$) Process Info - Exec Path** *doesn't contain* **outlook.exe**

    説明 このルールは、outlook.exe 以外のプロセスから電子メールファイル（.ost および .pst）にアクセスした場合にアラートを出し、記録します。

31. **名前** T1070 - ホストでのインジケータ削除 - イベントログ

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **wevtutil.exe and Process Info - Command String** *matches* **.*\s(cl|clear-log)\s.***

    説明 このルールは、wevtutil.exe を使用してイベントログが消去される場合にアラートを出し、記録します。

32. **名前** T1070 - ホストでのインジケータ削除 - USN

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **fsutil.exe and Process Info - Command String** *matches* **.*\susn\s.* and Process Info - Command String** *matches* **.*\sdeletejournal.***

    説明 このルールは、fsutil.exe を使用して USN ジャーナルが削除される場合にアラートを出し、記録します。

33. **名前** T1053 - スケジュールされたタスク

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **schtasks.exe and Process Info - Command String** *contains* **create**

    説明 このルールは、schtasks.exe を使用して新しいスケジュールされたタスクが作成される場合にアラートを出し、記録します。

34. **名前** T1003 - ログイン情報ダンピング - Vaultcmd

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **vaultcmd.exe and Process Info - Command String** *matches* **.*\/list.***

    説明 このルールは、vaultcmd.exe を使用して Windows Credentials Vault にアクセスされた場合にアラートを出し、記録します。

35. **名前** T1003 - ログイン情報ダンピング - レジストリ

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **reg.exe and ((Process Info - Command String** *contains* **save or Process Info - Command String** *contains* **export) and (Process Info - Command String** *contains* **hklm or Process Info - Command String** *contains* **hkey_local_machine) and (Process Info - Command String** *contains* **sam or Process Info - Command String** *contains* **security or Process Info - Command String** *contains* **system))**

説明 このルールは、reg.exeを使用して特定のレジストリハイブをダンプされた場合にアラートを出し、記録します。

36. **名前** T1201 - パスワードポリシー検出 1

Clause Event type = Follow Process and Process Info - Exec Path *contains* **chage and Process Info - Command String** *contains* **-l**

説明 このルールは、chageユーティリティを使用してLinuxマシンのパスワードポリシー（パスワード有効期間ポリシー）が一覧表示された場合にアラートを出し、記録します。

37. **名前** T1081 - ファイル内のログイン情報 - Linux

Clause Event type = Follow Process and (Process Info - Exec Path *contains* **cat or Process Info - Exec Path** *contains* **grep) and (Process Info - Command String** *contains* **.bash_history or Process Info - Command String** *contains* **.password or Process Info - Command String** *contains* **.passwd)**

説明 このルールは、Linux マシン上のファイルに保存されているパスワードを検索する試みが実行された場合にアラートを出し、記録します。

38. **名前** T1081 - ファイル内のログイン情報 - Windows

Clause Event type = Follow Process and Process Info - Exec Path *contains* **findstr.exe and Process Info - Command String** *contains* **password**

説明 このルールは、Windowsマシン上のファイルに保存されているパスワードを検索する試みが実行された場合にアラートを出し、記録します。

39. **名前** T1089 - セキュリティツールの無効化

Clause Event type = Follow Process and ((Process Info - Exec Path *contains* **fltmc.exe and Process Info - Command String** *contains* **unload sysmon) or (Process Info - Exec Path** *contains* **sysmon.exe and Process Info - Command String** *contains* **/u))**

説明 このルールは、fltmc.exe または sysmon.exe を使用して sysmon ドライバをアンロードする試みが実行された場合にアラートを出し、記録します。

フォレンジックプロファイル

プロファイルの追加

このセクションでは、新しいフォレンジックプロファイルを追加する方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

手順

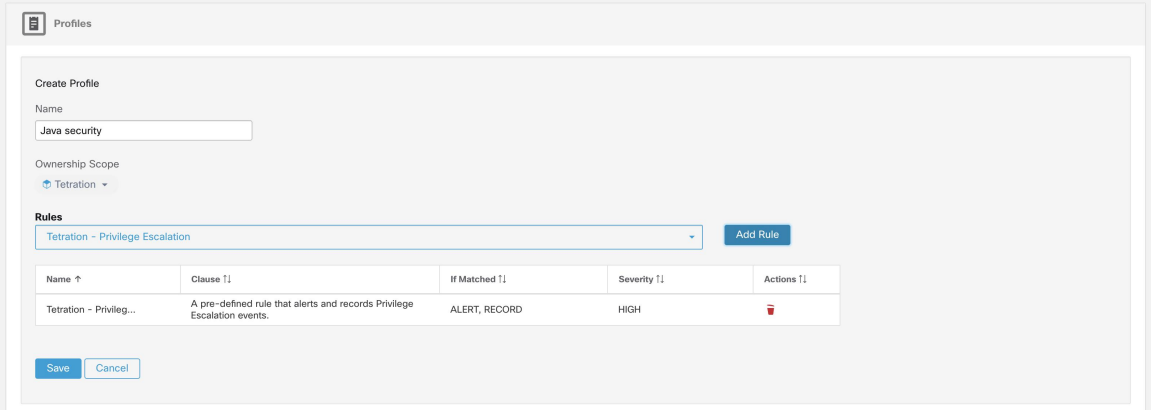
ステップ 1 左側のナビゲーションバーで、**[防御（Defend）] > [フォレンジックルール（Forensic Rules）]** をクリックします。

ステップ 2 [プロファイルの作成（Create Profile）] をクリックします。

ステップ 3 以下のフィールドに適切な値を入力します。

フィールド	説明
名前	プロファイル名を入力します名前は空白にできません。
[所有範囲（Ownership scope）]	このプロファイルの所有範囲を入力します。
ルール（Rule）	このプロファイルにルールを追加します。

図 *362 :* プロファイルの作成



ステップ 4 [保存（Save）] をクリックします。

プロファイルの編集

このセクションでは、ユーザーがフォレンジックプロファイルを編集する方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

手順

ステップ 1 左側のナビゲーションバーで、**[防御（Defend）] > [フォレンジックルール（Forensic Rules）]** をクリックします。

ステップ 2 編集するプロファイルを見つけて、右側の列にある**鉛筆**アイコンをクリックします。

ステップ 3 以下のフィールドに適切な値を入力します。

フィールド	説明
名前	プロファイルの名前を更新します。名前は空白にできません。
[所有範囲（Ownership scope）]	このプロファイルの所有範囲を更新します。
ルール（Rule）	このプロファイルにルールを追加/削除します。

ステップ 4 [保存（Save）] をクリックします。

プロファイルの複製

このセクションでは、ユーザーがフォレンジックプロファイルを複製する方法について説明します。

手順

ステップ 1 左側のナビゲーションバーで、**[防御（Defend）] > [フォレンジックルール（Forensic Rules）]** をクリックします。

ステップ 2 複製するプロファイルを見つけて、右側の列にある [複製（clone）] アイコンをクリックします。

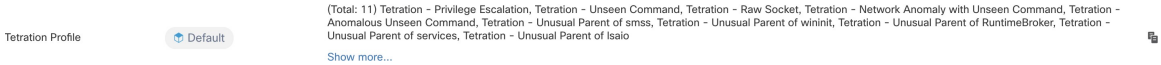
ステップ 3 複製されたプロファイルの名前を入力します。

ステップ 4 [保存（Save）] をクリックします。

デフォルトプロファイル：Secure Workload プロファイル

Secure Workload プロファイルには 11 のデフォルト フォレンジック ルールが含まれており、インテントに追加できます。ユーザーによる編集はできませんが、複製は可能です。複製されたデフォルト フォレンジック プロファイルは編集できます。

図 *363 :* デフォルトプロファイル



デフォルトのプロファイル：MITRE ATT&CK プロファイル

MITRE ATT&CK プロファイルには 39 の MITER ATT&CK ルールが含まれており、インテントに追加できます。ユーザーによる編集はできませんが、複製は可能です。複製されたプロファイルは編集できます。MITRE ATT&CK プロファイルには、次のルールが含まれます。

1. 疑わしい MS Office の動作
2. T1015 - アクセシビリティ機能 1
3. T1015 - アクセシビリティ機能 2
4. T1085 - rundll32
5. T1118 - InstallUtil
6. T1121 - Regsvcs/Regasm
7. T1127 - 信頼できる開発者ユーティリティ - msbuild.exe
8. T1127 - 信頼できる開発者ユーティリティ - rcsi.exe
9. T1127 - 信頼できる開発者ユーティリティ - tracker.exe
10. T1128 - Netsh ヘルパー Dll
11. T1136 - アカウントの作成
12. T1138 - アプリケーションシミング
13. T1180 - スクリーンセーバー
14. T1191 - CMSTP
15. T1202 - 間接コマンド実行 - forfiles.exe
16. T1202 - 間接コマンド実行 - pcalua.exe
17. T1216 - 署名付きスクリプトプロキシ実行 - pubprn.vbs
18. T1218 - 署名付きバイナリプロキシ実行 - msiexec.exe
19. T1218 - 署名付きバイナリプロキシ実行 - odbcconf.exe
20. T1218 - 署名付きバイナリプロキシ実行 - Register-CimProvider
21. T1220 - XSL スクリプト処理 - msxsl.exe
22. T1220 - XSL スクリプト処理 - wmic
23. T1223 - コンパイル済み HTML ファイル
24. T1003 - ログイン情報ダンピング - Lsass
25. T1140 - ファイルまたは情報の難読化解除/復号化
26. T1076 - リモート デスクトップ プロトコル

27. T1197 - BITS ジョブ - Powershell
28. T1170 - MSHTA
29. T1158 - 非表示ファイルおよびディレクトリ
30. T1114 - 電子メール収集
31. T1070 - ホストでのインジケータ削除 - イベント ログ
32. T1070 - ホストでのインジケータ削除 - USN
33. T1053 - スケジュールされたタスク
34. T1003 - ログイン情報ダンピング - Vaultcmd
35. T1003 - ログイン情報ダンピング - レジストリ
36. T1201 - パスワードポリシー検出 1
37. T1081 - ファイル内のログイン情報 - Linux
38. T1081 - ファイル内のログイン情報 - Windows
39. T1089 - セキュリティツールの無効化

変更ログ：フォレンジック

ルート範囲で SCOPE_OWNER 機能を持つ**サイト管理者**およびユーザーは、以下に示すアイコンをクリックすることで、各フォレンジック ルール、プロファイル、およびインテントの変更ログを表示できます。

図 *364 :* 変更ログ



これらのユーザーは、対応するテーブルの下にある [削除済みルール/プロファイル/インテントの表示（View Deleted Rules/Profiles/Intents）] リンクをクリックして、削除されたルール、プロファイル、およびインテントのリストを表示することもできます。

変更ログの詳細については、「ログの変更」を参照してください。ルート範囲の所有者は、その範囲に属するエンティティにおけるログの変更エントリの表示に制限されます。

法医学の可視化

フォレンジックページへのアクセス

このセクションでは、フォレンジックページにアクセスする方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

手順

ステップ 1 左側のパネルで [セキュリティ（Security）] リンクをクリックします。

ステップ 2 [フォレンジック（Forensics）]項目をクリックします。フォレンジックページが表示されます。

図 *365 :* セキュリティフォレンジック

フォレンジックイベントの閲覧

このセクションでは、一致するフォレンジックイベントを参照する方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインし、フォレンジックページに移動する必要があります。

手順

ステップ 1 ページ上部の [時間範囲ピッカー（Time Range Picker）] で特定の範囲を選択します。

ステップ 2 [重大度（Severity）] ドロップダウンを選択します。

ステップ 3 [フィルタ（Filters）] で、一致するフォレンジックイベントのフィルタを入力し、[フォレンジックイベントのフィルタ処理（Filter Forensic Events）] をクリックします。

ステップ 4 選択した時間範囲、重大度、およびフィルタに従って、一致するフォレンジックイベントのテーブルが更新されます。

（注） フォレンジックイベントはルート範囲レベルで表示され、サブ/子範囲に切り替えると表示されません。

フォレンジックイベントの検査

このセクションでは、フォレンジックイベントを検査する方法について説明します。

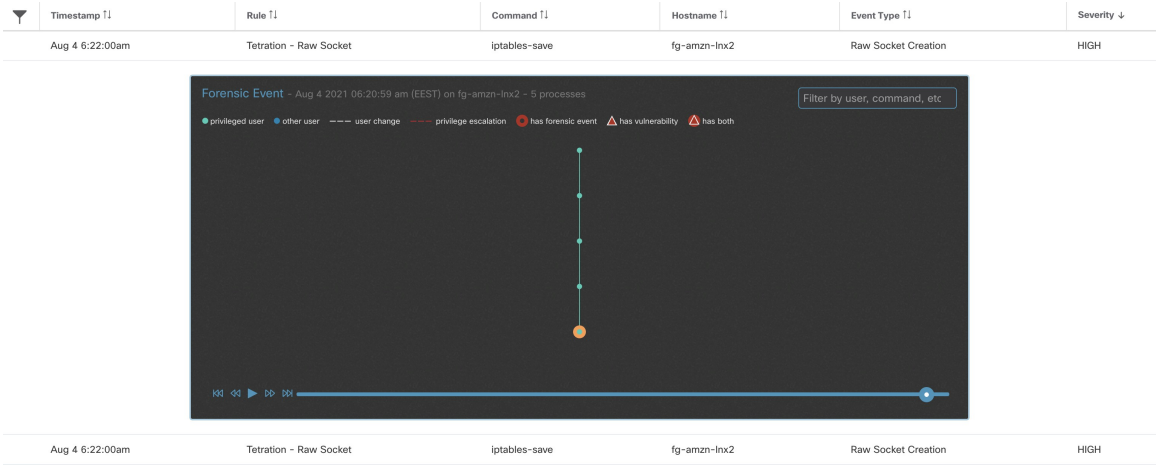
はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または[範囲所有者（ルート範囲）（Scope Owner（Root Scope））] としてシステムにログインする必要があります。

手順

ステップ 1 検査するイベントをクリックします。[プロセスの詳細（Process detail）] ペインが表示されます。

図 *366 :* フォレンジックイベントの表



ステップ 2 系統ツリーで、検査するプロセスをクリックして詳細を表示します。

図 *367 :* フォレンジックプロセスの詳細



フォレンジックイベントに表示されるフィールド

各フォレンジックイベントには、有用なデータを提供する多くのフィールドがあります。さまざまなタイプのフォレンジックイベントすべてに共通するフィールドがいくつかあります。また、特定のフォレンジックイベントに固有のフィールドもいくつかあります。

以下は、UI を構成するフィールドのリストです。最初の表では、すべてのフォレンジックイベントに共通するフィールドについて説明します。次に、各アラートとともに表示されるプロセス情報を説明する表、フォレンジックイベントごとに固有のフィールドを記載した表が続きます。データの保存方法やエクスポート方法が原因で、一部のフィールドは複数のテーブルに存在する場合があることに注意してください。

共通のフィールド

フィールド	説明
Bin attr ctime	Linux での変更時刻/Windows での作成時刻（バイナリ形式）
Bin attr hash	バイナリの SHA256 ハッシュ
Bin attr mtime	バイナリの変更時刻
Bin attr name	ファイルシステム上のバイナリの名前
Bin attr size	ファイルシステム上のバイナリのサイズ
Bin exec path	バイナリのフルパス
Cmdline	実行されるプロセスの完全なコマンドライン
Event time usec	このイベントが観測された時間（マイクロ秒単位）

Process Info

フィールド	説明
プロセス ID（Process ID）	プロセスのプロセス ID
[親プロセス ID（Parent Process ID）]	プロセスの親のプロセス ID
ユーザ	プロセスを実行したユーザー
実行パス（Execution path）	プロセスに対応するバイナリのフルパス。
開始時刻（Start time）	プロセスが開始された時刻。
フルコマンド（Full command）	実行されるプロセスの完全なコマンドライン

特権昇格

フィールド	説明
親コマンドライン（Parent cmdline）	プロセスの親の完全なコマンドライン
Parent exe	プロセスの親のフルパス
親の稼働時間（マイクロ秒）	プロセスの親が実行されてからの時間

フィールド	説明
親ユーザー名（Parent Username）	プロセスの親を実行したユーザー
Types bitmap suid binary	バイナリに suid ビットが設定されているかどうかを表示

ユーザーログオン

フィールド	説明
[認証タイプ - パスワード（Auth type password）]	パスワード認証を示します
[認証タイプ - 公開キー（Auth type pubkey）]	キーベースの認証を示します
[ログインタイプ - SSH（Type login ssh）]	ユーザーが SSH 経由でログインしたことを示します
[ログインタイプ - バッチ（Type login win batch）]	Windows バッチログインを示します（タイプ 4、たとえば schtasks）
[ログインタイプ - キャッシュ（Type login win cached）]	キャッシュされた資格情報によるログオンを示します（タイプ 11、CachedIntetractive）
[ログインタイプ - インタラクティブ（Type login win interactive）]	インタラクティブログオンを示します（タイプ 2、RDP など）
[ログインタイプ - ネットワーククリアテキスト（Type login win network cleartext）]	SSH 経由のログオンを示します（タイプ 8）
[ログインタイプ - ネットワーク（Type login win network）]	ネットワークログインを示します（タイプ3、Psexec など）
[ログインタイプ-新しい資格情報の使用（Type login win new cred）]	新しい資格情報の使用を示します（タイプ9、Runas コマンドなど）
[ログインタイプ - リモートインタラクティブ（Type login win remote interactive）]	リモートログオンを示します（タイプ10、RDP など）
[ログインタイプ - サービス（Type login win service）]	サービスがSCMによって開始されたことを示します（タイプ 5）
[ログインタイプ - ロック解除（Type login win unlock）]	ワークステーションがロック解除されたことを示します（タイプ 7）
[送信元IP（Src IP）]	ログインイベントが生成されたソース IP
[送信元ポート（Src Port）]	ログインイベントが生成されたソースポート

フィールド	説明
[ユーザー名（Username）]	ログインイベントに関連付けられているユーザー名

ユーザーログオン失敗

フィールド	説明
[認証タイプ - パスワード（Auth type password）]	パスワード認証を示します
[認証タイプ - 公開キー（Auth type pubkey）]	キーベースの認証を示します
[ログインタイプ - SSH（Type login ssh）]	ユーザーが SSH 経由でログインしたことを示します
[ログインタイプ - バッチ（Type login win batch）]	Windows バッチログインを示します（タイプ 4、たとえば schtasks）
[ログインタイプ - キャッシュ（Type login win cached）]	キャッシュされた資格情報によるログオンを示します（タイプ 11、CachedIntetractive）
[ログインタイプ - インタラクティブ（Type login win interactive）]	インタラクティブログオンを示します（タイプ 2、RDP など）
[ログインタイプ - ネットワーククリアテキスト（Type login win network cleartext）]	SSH 経由のログオンを示します（タイプ 8）
[ログインタイプ - ネットワーク（Type login win network）]	ネットワークログインを示します（タイプ 3、Psexec など）
[ログインタイプ -新しい資格情報の使用（Type login win new cred）]	新しい資格情報の使用を示します（タイプ 9、Runas コマンドなど）
[ログインタイプ - リモートインタラクティブ（Type login win remote interactive）]	リモートログオンを示します（タイプ 10、RDP など）
[ログインタイプ - サービス（Type login win service）]	サービスが SCM によって開始されたことを示します（タイプ 5）
[ログインタイプ - ロック解除（Type login win unlock）]	ワークステーションがロック解除されたことを示します（タイプ 7）
[送信元IP（Src IP）]	ログインイベントが生成されたソース IP
[送信元ポート（Src Port）]	ログインイベントが生成されたソースポート

フィールド	説明
[ユーザー名（Username）]	ログインイベントに関連付けられているユーザー名

シェルコード

フィールド	説明
Signal sources bitmap cmd as sh no tty	シェルプロセスに関連付けられたttyがないことを示します。
Signal sources bitmap powershell	プロセスにpowershell dllがロードされていることを示します（System.Management.Automation）。

File Access

フィールド	説明
ファイル（File）	アクセスしたファイルのフルパス
Perm read perm	ファイルに読み取り権限があることを示します
Perm read write perm	ファイルに読み取りおよび書き込み権限があることを示します
Perm write perm	ファイルに書き込み権限があることを示します

ユーザーアカウント

フィールド	説明
[ユーザー名（Username）]	作成されたユーザーのユーザー名
[操作アカウントの追加（Ops acct add）]	新しいアカウントが追加されたことを示します

未確認コマンド

フィールド	説明
異常 - スコア（Anomaly - Score）	コマンドラインが以前に表示された頻度を示すスコア（0 ～ 1.0）。スコアが低いほど、コマンドがより異常であることを意味します
異常 - 類似性 - 高（Anomaly - Similarity - High）	異常スコアが 0.8 より大きく、1 より小さい場合は true
異常 - 類似性 - 中（Anomaly - Similarity - Medium）	異常スコアが 0.6 より大きく、0.8 以下の場合は true
異常 - 類似性 - 低（Anomaly - Similarity - Low）	異常スコアが 0 より大きく、0.6 以下の場合は true
異常 - 類似 - 確認済み（Anomaly - Similarity - Seen）	異常スコアが 1 の場合、つまり、同じコマンドが以前に見られた場合は true
異常 - 類似性 - ユニーク （Anomaly - Similarity - Unique）	異常スコアが 0 の場合、つまり、コマンドがこれまでに見られなかった場合は true
親コマンドライン（Parent cmdline）	親プロセスの完全なコマンドライン
親 ExePath（Parent exepath）	親プロセスのバイナリパス
親の稼働時間（Parent uptime）	親プロセスが実行されてからの時間
親ユーザー名（Parent username）	親プロセスを実行したユーザーのユーザー名
センサー稼働時間（Sensor uptime）	センサーの稼働時間

未確認ライブラリ

フィールド	説明
ライブラリパス（Lib Path）	以前はプロセスに関連付けられていなかったライブラリファイルの完全パス

raw ソケットの作成

フィールド	説明
Exe Path	raw ソケットを作成したプロセスの完全パス

ライブラリ変更

フィールド	説明
ライブラリの変更された名前	変更されたライブラリのフルパス

サイドチャネル

フィールド	説明
信号ソースビットマップ Meltdown	Meltdown エクスプロイトの使用を示します。

ユーザーログオンの追跡

フィールド	説明
[ユーザー名（Username）]	プロセスを実行したユーザー名

移行手順

フィールド	説明
親コマンドライン（Parent cmdline）	親プロセスの完全なコマンドライン
親 ExePath（Parent exepath）	親プロセスのバイナリパス
親の稼働時間 usec（Parent uptime usec）	親プロセスが実行されてからの時間
親ユーザー名（Parent username）	親プロセスを実行したユーザーのユーザー名
usec が最後に変更されてからの時間	プロセス開始時刻からバイナリファイル変更時刻までの経過時間（mtime）
ユーザー名	プロセスを実行したユーザーのユーザー名

ネットワーク異常

ネットワーク異常イベントに関連する属性のリストについては、ネットワーク異常イベントのフォレンジックルールを参照してください。

フォレンジック分析：検索可能なフィールド

以下の表は、[フォレンジック分析（Forensics Analysis）]ページの検索バーで検索可能なフィールドの説明を示しています。

その他のフィールド

フィールド	説明
フォレンジックルール名（Forensic Rule Name）	特定のフォレンジックルールによってラベル付けされたイベント
Hostname	特定のホスト名からのイベント
センサー ID（Sensor ID）	特定のセンサーからのイベント
重大度	特定の重大度のイベント

フォレンジック分析の検索用語

共通のフィールド

これらのフィールドは、さまざまなイベントタイプで共通しています。「Event name - Event」というプレフィックスが付いています。たとえば、「Binary Changed - Binary Attribute - CTime (epoch nanoseconds)」などです。

フィールド	説明
Binary Attribute - CTime (epoch nanoseconds)	Linux での変更時刻/Windows での作成時刻（バイナリ形式）
Binary Attribute - Hash	バイナリの SHA256 ハッシュ
Binary Attribute - MTime (epoch nanoseconds)	バイナリの変更時刻
Binary Attribute - Filename	ファイルシステム上のバイナリの名前
Binary Attribute - Size (bytes)	ファイルシステム上のバイナリのサイズ
Event Binary Path	バイナリのフルパス
コマンド ライン	実行されるプロセスの完全なコマンドライン

バイナリ変更

「共通フィールド」の表に記載されている以外に検索用語はありません。

File Access

ファイルアクセスの検索用語には、「File Access -」というプレフィックスが付いています（例：「File Access - Filename」）。

フィールド	説明
ファイル名	アクセスしたファイルのフルパス
Is = Permission - Read	ファイルに読み取り権限があることを示します
Is = Permission - ReadWrite	ファイルに読み取りおよび書き込み権限があることを示します
Is = Permission - Write	ファイルに書き込み権限があることを示します

移行手順

プロセス後検索語には、プレフィックス「Follow Process -」が付いています（例：「Follow Process - Parent Command Line」）。

フィールド	説明
親コマンドライン（Parent Command Line）	親プロセスの完全なコマンドライン
親 Exec パス（Parent Exec Path）	親プロセスのバイナリパス
親の稼働時間（Parent Uptime）（マイクロ秒）	親プロセスが実行されてからの時間
親ユーザー名（Parent Username）	親プロセスを実行したユーザーのユーザー名
最後にファイルが変更されてからのプロセス開始時間（Process Start Time Since Last File Changed）（マイクロ秒）	プロセスの開始から最新の（対応する）ファイル変更までの経過時間
ユーザー名	後続するプロセスに関連付けられたユーザー名

ユーザーログオンの追跡

Follow User Logon 検索用語には、「Follow User Logon -」というプレフィックスが付いています（例：「Follow User Logon - Username」）。

フィールド	説明
[ユーザー名（Username）]	プロセスに関連付けられたユーザー名

Ldap

Ldap 検索語には、プレフィックス「Ldap -」が付いています（例：「Ldap - Department」）。

フィールド	説明
部署名（Department）	プロセスのユーザー名に関連付けられた AMS Ldap ユーザーの部門（利用可能な場合）
説明	プロセスのユーザー名に関連付けられた AMS Ldap ユーザーの説明（利用可能な場合）
ユーザー名	プロセスに関連付けられた AMS Ldap ユーザー名（利用可能な場合）

ライブラリ変更

Library Changed 検索用語には、「Library Changed -」というプレフィックスが付きます。例：「Library Changed - Department」

フィールド	説明
ライブラリファイル名（Lib Filename）	変更されたライブラリのフルパス

特権昇格

Privilege Escalation 検索用語には、プレフィックス「Privilege Escalation -」が付いています（例：「Privilege Escalation - Parent Command Line」）。

フィールド	説明
親コマンドライン（Parent Command Line）	プロセスの親の完全なコマンドライン
親 Exec パス	プロセスの親のフルパス
親の稼働時間（マイクロ秒）	プロセスの親が実行されてからの時間
親ユーザー名（Parent Username）	プロセスの親を実行したユーザー

フィールド	説明
タイプ：SUID バイナリ	バイナリに suid ビットが設定されているかどうかを表示

Process Info

Process Info 検索語には、接頭辞「Process Info - 」が付いています（例：「Process Info - Binary Hash」）。

フィールド	説明
バイナリハッシュ（Binary Hash）	プロセスに関連付けられたバイナリのハッシュ
トークン化されたコマンド文字列（Command String Tokenized）	プロセスのトークン化されたコマンドライン。
コマンド文字列	プロセスの完全なコマンドライン
Exec Path	プロセスに対応するバイナリのフルパス。

Raw Socket

raw ソケット検索語には、プレフィックス「Raw Socket -」が付いています （例：「Raw Socket - Exec Path」）。

フィールド	説明
Exec Path	raw ソケットを作成したプロセスの完全パス

シェルコード

シェルコードの検索用語には、「Shellcode -」のプレフィックスが付いています（例：「Shellcode - Source - Not From Login」）。

フィールド	説明
ソース - ログインからではない（Source - Not From Login）	シェルプロセスに関連付けられた tty がないことを示します。
ソース - Powershell（Source - Powershell）	プロセスに powershell dll がロードされていることを示します（System.Management.Automation）。

サイドチャネル

サイドチャネルの検索語には、プレフィックス「シェルコード -（Shellcode ）」が付いています。たとえば、[シェルコード - ソース - Meltdown（Shellcode - Source - Meltdown）] となります。

フィールド	説明
[ソース - Meltdown（Source - Meltdown）]	Meltdown エクスプロイトの使用を示します。

未確認コマンド

未確認コマンドの検索用語には、「Unseen Command -」というプレフィックスが付きます。例：Unseen Command - Anomaly - Similarity - High

フィールド	説明
異常 - スコア（Anomaly - Score）	コマンドラインが以前に表示された頻度を示すスコア（0 ～ 1.0）。スコアが低いほど、コマンドがより異常であることを意味します
異常 - 類似性 - 高（Anomaly - Similarity - High）	異常スコアが 0.8 より大きく、1 より小さい場合は true
異常 - 類似性 - 中（Anomaly - Similarity - Medium）	異常スコアが 0.6 より大きく、0.8 以下の場合は true
異常 - 類似性 - 低（Anomaly - Similarity - Low）	異常スコアが 0 より大きく、0.6 以下の場合は true
異常 - 類似 - 確認済み（Anomaly - Similarity - Seen）	異常スコアが 1 の場合、つまり、同じコマンドが以前に見られた場合は true
異常 - 類似性 - ユニーク （Anomaly - Similarity - Unique）	異常スコアが 0 の場合、つまり、コマンドがこれまでに見られなかった場合は true
親コマンドライン（Parent cmdline）	親プロセスの完全なコマンドライン
親 ExePath（Parent Exepath）	親プロセスのバイナリパス
親の稼働時間（Parent uptime）	親プロセスが実行されてからの時間
親ユーザー名 （Parent Username）	親プロセスを実行したユーザーのユーザー名
センサー稼働時間（Sensor Uptime）	センサーの稼働時間
異常 - 最新の類似コマンド（Anomaly - Latest Similar Commands）	イベントのコマンドに類似した、以前に観察された最新の 5 つのコマンド

未確認ライブラリ

未確認ライブラリの検索用語には、「Unseen Library -」というプレフィックスが付いています（例：「Unseen Library - Lib Filename」）。

フィールド	説明
ライブラリファイル名（Lib Filename）	以前はプロセスに関連付けられていなかったライブラリファイルの完全パス

ユーザーアカウント

ユーザーアカウントの検索語には、プレフィックス「ユーザーアカウント -」が付いています（例：「ユーザー アカウント - アカウント名」）。

フィールド	説明
アカウント名	作成されたユーザーのユーザー名
操作 - アカウントの追加	新しいアカウントが追加されたことを示します

ユーザーログオン

ユーザーログオンの検索用語には、プレフィックス「ユーザーログオン -（User Logon -）」が付いています。例：「ユーザーログオン - 認証タイプ - パスワード（User Logon - Auth Type - Password）」

フィールド	説明
[認証タイプ - パスワード（Auth Type - Password）]	パスワード認証を示します
[認証タイプ - 公開キー（Auth type - Pubkey）]	キーベースの認証を示します
[ログインタイプ - SSH経由のログイン（Login Type - Login Via SSH）]	ユーザーが SSH 経由でログインしたことを示します
[ログインタイプ - Windowsバッチログイン（Login Type - Windows Login Batch）]	Windows バッチログインを示します（タイプ 4、schtasks など）
[ログインタイプ - Windowsキャッシュログイン（Login Type - Windows Login Cached）]	キャッシュされたクレデンシャルによるログオンを示します (タイプ 11、CachedIntetractive）

フィールド	説明
[ログインタイプ - Windowsインタラクティブログイン（Login Type - Windows Login Interactive）]	インタラクティブログオンを示します（タイプ 2、RDP など）
[ログインタイプ - Windowsネットワーククリアテキスト（Login Type - Windows Network Cleartext）]	SSH 経由のログオンを示します（タイプ 8）
[ログインタイプ - Windowsネットワーク（Login Type - Windows Network）]	ネットワークログインを示します（タイプ3、Psexec など）
[ログインタイプ - 新しいクレデンシャルによるWindowsログイン（Login Type - Windows Login New Credential）]	新しいクレデンシャルの使用を示します（タイプ 9、Runas コマンドなど）
[ログインタイプ - リモートインタラクティブWindowsログイン（Login Type - Windows Login Remote Interactive）]	リモートログオンを示します（タイプ10、RDP など）
[ログインタイプ - Windowsログインサービス（Login Type - Windows Login Service）]	サービスが SCM（タイプ 5）によって開始されたことを示します
[ログインタイプ - Windowsログインのロック解除（Login Type - Windows Login Unlock）]	ワークステーションがロック解除されたことを示します（タイプ 7）
[送信元IP（Source IP）]	ログインイベントが生成された送信元 IP
[送信元ポート（Source Port）]	ログインイベントが生成されたソースポート
[ユーザー名（Username）]	ログインイベントに関連付けられているユーザー名

ユーザーログオン失敗

User Logon Failed 検索語には、プレフィックス「User Logon Failed -」が付いています。例：「User Logon Failed - Auth Type - Password」

フィールド	説明
[認証タイプ - パスワード（Auth Type - Password）]	パスワード認証を示します
[認証タイプ - 公開キー（Auth type - Pubkey）]	キーベースの認証を示します
[ログインタイプ - SSH経由のログイン（Login Type - Login Via SSH）]	ユーザーが SSH 経由でログインしたことを示します

フィールド	説明
[ログインタイプ - Windowsバッチログイン（Login Type - Windows Login Batch）]	Windows バッチログインを示します（タイプ 4、schtasks など）
[ログインタイプ - Windowsキャッシュログイン（Login Type - Windows Login Cached）]	キャッシュされたクレデンシャルによるログオンを示します (タイプ 11、CachedIntetractive）
[ログインタイプ - Windowsインタラクティブログイン（Login Type - Windows Login Interactive）]	インタラクティブログオンを示します（タイプ 2、RDP など）
[ログインタイプ - Windowsネットワーククリアテキスト（Login Type - Windows Network Cleartext）]	SSH 経由のログオンを示します（タイプ 8）
[ログインタイプ - Windowsネットワーク（Login Type - Windows Network）]	ネットワークログインを示します（タイプ3、Psexec など）
[ログインタイプ - 新しいクレデンシャルによるWindowsログイン（Login Type - Windows Login New Credential）]	新しいクレデンシャルの使用を示します（タイプ 9、Runas コマンドなど）
[ログインタイプ - リモートインタラクティブWindowsログイン（Login Type - Windows Login Remote Interactive）]	リモートログオンを示します（タイプ10、RDP など）
[ログインタイプ - Windowsログインサービス（Login Type - Windows Login Service）]	サービスが SCM（タイプ 5）によって開始されたことを示します
[ログインタイプ - Windowsログインのロック解除（Login Type - Windows Login Unlock）]	ワークステーションがロック解除されたことを示します（タイプ 7）
[送信元IP（Source IP）]	ログインイベントが生成された送信元 IP
[送信元ポート（Source Port）]	ログインイベントが生成されたソースポート
[ユーザー名（Username）]	ログインイベントに関連付けられているユーザー名

フォレンジックアラート

フォレンジックイベントは、一致するルールに**アラート**アクションが含まれている場合、Secure Workload アラートシステムで見つけることができます。

フォレンジックアラートへのアクセス

このセクションでは、フォレンジックアラートにアクセスする方法について説明します。

はじめる前に

- [サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。
- [フォレンジック（Forensics）] アラート送信元のアラートをオンにする必要があります。

手順

ステップ 1 左側のツールバーの [アラート（Alerts）] をクリックします。

ステップ 2 [アラート（Alerts）] ページが表示されます。

アラート詳細の確認

はじめる前に

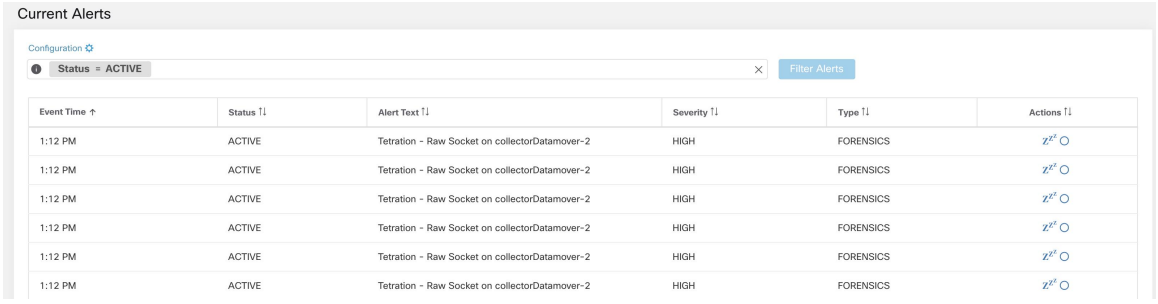
[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

手順

ステップ 1 アラートページから、確認するアラートをクリックします。

ステップ 2 プロファイル/ルールをクリックして、一致するフォレンジックプロファイル/ルールの詳細を表示します。アラートが発生した後に一致するプロファイル/ルールが更新されると、警告インジケータが表示されることに注意してください。

図 *368 :* フォレンジックアラートのページ



さらに、アラートをスヌーズしたり、含めたり除外したりすることができます。詳細については、「現在のアラート」セクションを参照してください。

外部との統合

フォレンジックアラートは、syslog などの外部監視ツールに送信できます。フォレンジックアラートは JSON 形式で送信されます。JSON フィールドの定義は、上記の「フォレンジックイベントに表示されるフィールド」セクションで定義されています。

JSON Kafka 出力の例は次のとおりです。


```
  {
    "severity": "HIGH",
    "tenant_id": 0,
    "alert_time": 1595573847156,
    "alert_text": "Tetration - Anomalous Unseen Command on collectorDatamover-1",
    "key_id":
"d89f926cddc7577553eb8954e492528433b2d08e:5efcfdf5497d4f474f1707c2:5efcfdf6497d4f474f1707d6:20196:CMD_NOT_SEEN",

    "alert_id": "/Alerts/5efcfdf5497d4f474f1707c2/DataSource{location_type='TETRATION',
 location_name='forensics', location_grain='MIN',
root_scope_id='5efcfdf5497d4f474f1707c2'}/db10d21631eebefc3b8d3aeaba5a0b1b45f4259e85b591763d7eaee9161ca076",

    "root_scope_id": "5efcfdf5497d4f474f1707c2",
    "type": "FORENSICS",
    "event_time": 1595573795135,
    "alert_details": "{\"Sensor
Id\":\"d89f926cddc7577553eb8954e492528433b2d08e\",\"Hostname\":\"collectorDatamover-1\",\"Process
 Id\":20196,\"scope_id\":\"5efcfdf5497d4f474f1707c2\",\"forensic\":{\"Unseen
Command\":\"true\",\"Unseen Command - Sensor Uptime
(microseconds)\":\"34441125356\",\"Unseen Command - Parent Uptime
(microseconds)\":\"35968418683\",\"Unseen Command - Parent Username\":\"root\",\"Unseen
 Command - Parent Command Line\":\"svlogd -tt /local/logs/tetration/efe/ \",\"Unseen
Command - Parent Exec Path\":\"/sbin/svlogd\",\"Unseen Command - Anomaly -
Score\":\"0\",\"Unseen Command - Anomaly - Similarity - Unique\":\"true\",\"Process Info
 - Command String\":\"gzip \",\"Process Info - Exec
Path\":\"/bin/gzip\"},\"profile\":{\"id\":\"5efcfdf6497d4f474f1707e4\",\"name\":\"Tetration

Profile\",\"created_at\":1586889,\"updated_at\":1586889,\"root_app_scope_id\":\"5efcfdf5497d4f474f1707c2\",\"rule\":{\"id\":\"5efcfdf6497d4f474f1707d6\",\"name\":\"Tetration
 - Anomalous Unseen
Command\",\"clause_chips\":\"[{\\\"type\\\":\\\"filter\\\",\\\"facet\\\":{\\\"field\\\":\\\"event_type\\\",\\\"title\\\":\\\"Event

type\\\",\\\"type\\\":\\\"STRING\\\"},\\\"operator\\\":{\\\"label\\\":\\\"\\u003d\\\",\\\"type\\\":\\\"eq\\\"},\\\"displayValue\\\":\\\"Unseen
 Command\\\",\\\"value\\\":\\\"Unseen
Command\\\"},{\\\"type\\\":\\\"filter\\\",\\\"facet\\\":{\\\"field\\\":\\\"forensic_event__cmd_not_seen_data__cmdline_anomaly_info_score\\\",\\\"title\\\":\\\"Unseen
 Command - Anomaly -
Score\\\",\\\"type\\\":\\\"NUMBER\\\",\\\"parent\\\":{\\\"label\\\":\\\"Data\\\",\\\"type\\\":\\\"1\\\",\\\"displayValue\\\":\\\"Data\\\",\\\"value\\\":\\\"Data\\\"}},\"created_at\":1586889,\"updated_at\":1586889,\"root_app_scope_id\":\"5efcfdf5497d4f474f1707c2\"}"

  }
```


alert_details の値自体がエスケープされた JSON 文字列であり、上記のアラートの内容を以下に示します。


```
  {
    "Sensor Id": "d89f926cddc7577553eb8954e492528433b2d08e",
```

```
    "Hostname": "collectorDatamover-1",
    "Process Id": 20196,
    "scope_id": "5efcfdf5497d4f474f1707c2",
    "forensic": {
      "Unseen Command": "true",
      "Unseen Command - Sensor Uptime (microseconds)": "34441125356",
      "Unseen Command - Parent Uptime (microseconds)": "35968418683",
      "Unseen Command - Parent Username": "root",
      "Unseen Command - Parent Command Line": "svlogd -tt /local/logs/tetration/efe/ ",
      "Unseen Command - Parent Exec Path": "/sbin/svlogd",
      "Unseen Command - Anomaly - Score": "0",
      "Unseen Command - Anomaly - Similarity - Unique": "true",
      "Process Info - Command String": "gzip ",
      "Process Info - Exec Path": "/bin/gzip"
    },
    "profile": {
      "id": "5efcfdf6497d4f474f1707e4",
      "name": "Tetration Profile",
      "created_at": 1593638390,
      "updated_at": 1593638390,
      "root_app_scope_id": "5efcfdf5497d4f474f1707c2"
    },
    "rule": {
      "id": "5efcfdf6497d4f474f1707d6",
      "name": "Tetration - Anomalous Unseen Command",
      "clause_chips":
"[{\"type\":\"filter\",\"facet\":{\"field\":\"event_type\",\"title\":\"Event
type\",\"type\":\"STRING\"},\"operator\":{\"label\":\"=\",\"type\":\"eq\"},\"displayValue\":\"Unseen
 Command\",\"value\":\"Unseen
Command\"},{\"type\":\"filter\",\"facet\":{\"field\":\"forensic_event__cmd_not_seen_data__cmdline_anomaly_info__score\",\"title\":\"Unseen
 Command - Anomaly -
Score\",\"type\":\"NUMBER\"},\"operator\":{\"label\":\"<\",\"type\":\"lt\"},\"displayValue\":\"0.6\",\"value\":\"0.6\"}]",

      "created_at": 1593638390,
      "updated_at": 1595539498,
      "root_app_scope_id": "5efcfdf5497d4f474f1707c2"
    }
 }
```


フォレンジックイベントの詳細は、フィールドフォレンジックに含まれます。フォレンジックイベントの属性のリストについては、「フォレンジックイベントに表示されるフィールド」を参照してください。フォレンジックイベントの属性は、UI のアラートの詳細にも表示されます。

フォレンジックスコア

フォレンジックスコアが表示される場所

セキュリティダッシュボード：

図 *369* : セキュリティダッシュボードの *[*フォレンジックスコア（***Forensics Score***）*]* セクション

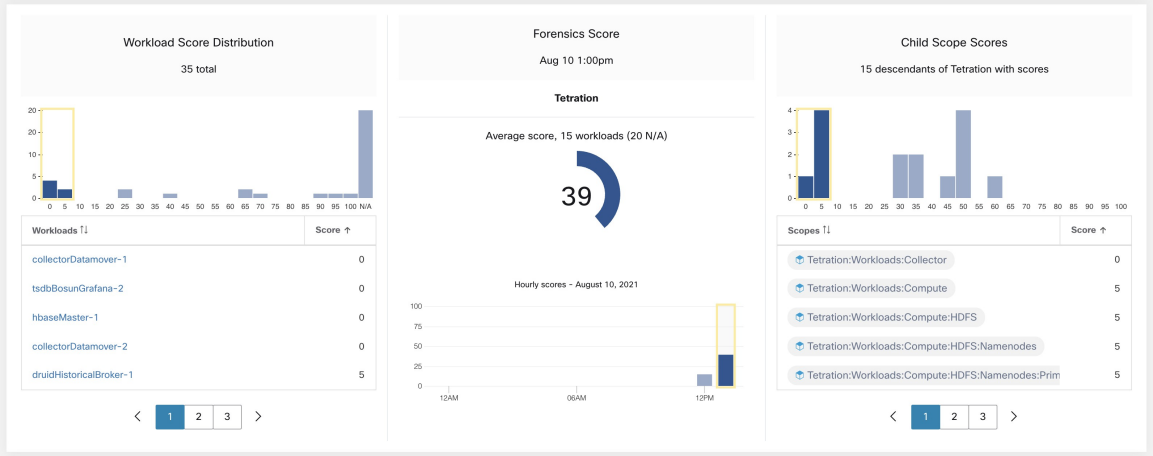
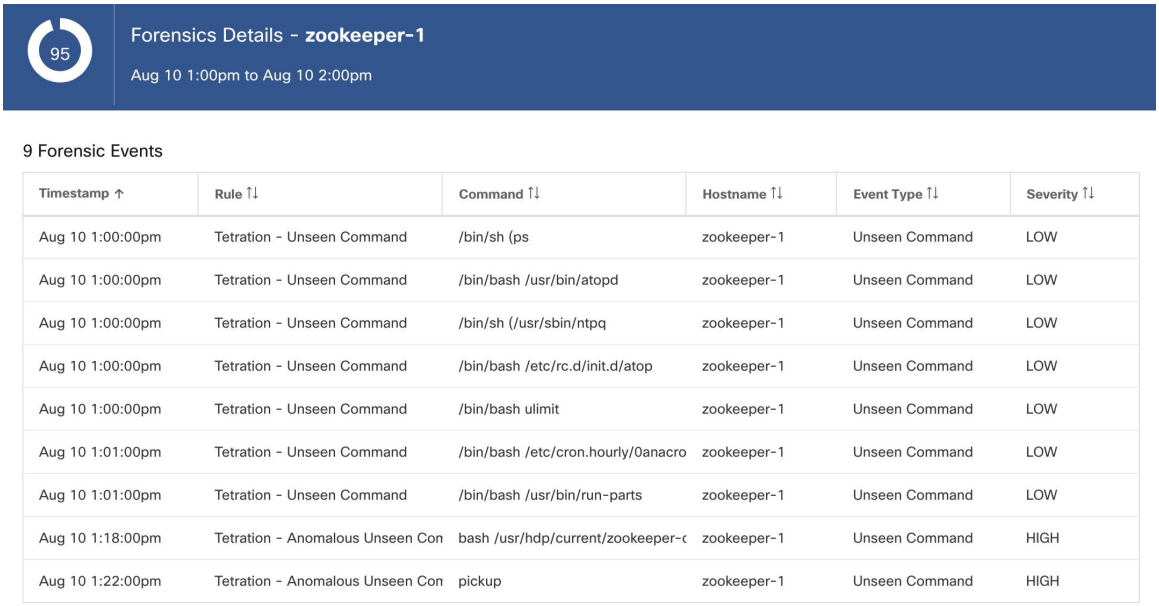


図 *370* : セキュリティダッシュボードの *[*フォレンジックスコアの詳細（***Forensics Score Details***）*]* セクション



Timestamp	Rule	Command	Hostname	Event Type	Severity
Aug 10 1:00:00pm	Tetration - Unseen Command	/bin/sh (ps	zookeeper-1	Unseen Command	LOW
Aug 10 1:00:00pm	Tetration - Unseen Command	/bin/bash /usr/bin/atopd	zookeeper-1	Unseen Command	LOW
Aug 10 1:00:00pm	Tetration - Unseen Command	/bin/sh (/usr/sbin/ntpq	zookeeper-1	Unseen Command	LOW
Aug 10 1:00:00pm	Tetration - Unseen Command	/bin/bash /etc/rc.d/init.d/atop	zookeeper-1	Unseen Command	LOW
Aug 10 1:00:00pm	Tetration - Unseen Command	/bin/bash ulimit	zookeeper-1	Unseen Command	LOW
Aug 10 1:01:00pm	Tetration - Unseen Command	/bin/bash /etc/cron.hourly/0anacro	zookeeper-1	Unseen Command	LOW
Aug 10 1:01:00pm	Tetration - Unseen Command	/bin/bash /usr/bin/run-parts	zookeeper-1	Unseen Command	LOW
Aug 10 1:18:00pm	Tetration - Anomalous Unseen Con	bash /usr/hdp/current/zookeeper-c	zookeeper-1	Unseen Command	HIGH
Aug 10 1:22:00pm	Tetration - Anomalous Unseen Con	pickup	zookeeper-1	Unseen Command	HIGH

フォレンジックスコアの計算方法

フォレンジックスコアはワークロードごとに計算されます。ワークロードのフォレンジックスコアは、該当範囲で有効になっているプロファイルに基づいて、そのワークロードで観察されたフォレンジックイベントから得られます。スコア 100 は、有効なプロファイルの設定されたルールを介してフォレンジックイベントが観察されなかったことを意味し、スコア 0 は、即時対応が必要なフォレンジックイベントが検出されたことを意味します。ある範囲のフォレンジックスコアは、その範囲内の平均ワークロードスコアです。特定の時間のフォレンジックスコアは、その時間内のすべてのスコアの最小値です。

- 重大度が REQUIRES IMMEDIATE ACTION のフォレンジックイベントは、範囲全体のスコアをゼロに減らします。
- 重大度が CRITICAL のフォレンジックイベントは、ワークロードのスコアを 10 の重みで減らします。
- 重大度が HIGH のフォレンジックイベントは、ワークロードのスコアを重み 5 で減らします。
- 重大度が MEDIUM のフォレンジックイベントは、ワークロードのスコアを重み 3 で減らします。
- 重大度が LOW のフォレンジックイベントは、フォレンジックスコアに寄与しません。これは、信号の品質がまだ調整中であり、ノイズが多い可能性がある新しいルールに推奨されます。

たとえば、重大度が CRITICAL の 2 つのルール、重大度が HIGH の 1 つのルール、重大度が LOW の 1 つのルールにそれぞれ一致する 3 つのフォレンジックイベントがワークロードにあるとします。そのワークロードのフォレンジックスコアは、100 - 1 * 10 - 1 * 5 - 1 * 0 = 85 です。

フォレンジックスコアは、フォレンジック機能が有効になっていないワークロードの場合は該当しません。

フォレンジックスコアの改善方法

フォレンジックスコアの調整は、有効なフォレンジックルールを調整して行います。ノイズの少ないルールを作成すると、より正確なスコアが得られます。正当なフォレンジックイベント（侵入またはその他の悪質なアクティビティの証拠となるイベント）に対応して防止するのも、フォレンジックスコアを改善するための良い方法です。

警告

- フォレンジックスコアの詳細には、その時間内のすべてのフォレンジックイベントが表示されます。つまり、フォレンジックスコアの詳細には、フォレンジックスコアの計算に使用されるもの以外のフォレンジックイベントが表示される場合があります。
- フォレンジックスコアは現在、優れた可視性センサーと適用センサーで利用できます。

PCR ベースのネットワーク異常検出

ネットワークの異常機能は、Producer Consumer Ratio（PCR）の概念に基づいて、ワークロードに出入りする異常に大量のデータを検出します。PCR は次のように定義されます。

```
Egress app byte count - Ingress app byte count
  PCR = ------------------------------------------
        Egress app byte count + Ingress app byte count
```

PCR の値は [-1.0, 1.0] の範囲にあります。ここで、

- PCR = 1.0 は、ワークロードが純粋にデータを送信することを意味します
- PCR = -1.0 は、ワークロードが純粋にデータを受信することを意味します
- PCR = 0.0 は、ワークロードでデータの送受信のバランスが取れていることを意味します

他のフォレンジック機能と同様に、インテントベースの設定を使用して、記録したり警告したりするネットワーク異常イベントを設定できます。ワークロードから検出されたネットワーク異常イベントは5分ごとにエクスポートされ、設定されたルールと5分後に照合されます。その結果、新しいネットワーク異常イベントは、イベントの時刻から最大10分の遅延で、5分ごとにUIでのみ観測されます。

（注） Secure Workload ソフトウェアの 3.2 および 3.1 バージョンでは、ネットワーク異常検出はデータリーク検出と呼ばれていました。

ネットワーク異常イベントのフォレンジックルール

フォレンジックルールを追加する方法については、「フォレンジック設定」を参照してください。

ルールの属性

このセクションでは、ネットワーク異常関連のルールを定義する属性の詳細について説明します。最も単純なネットワーク異常ルールは次のとおりです。

Event Type = Network Anomaly

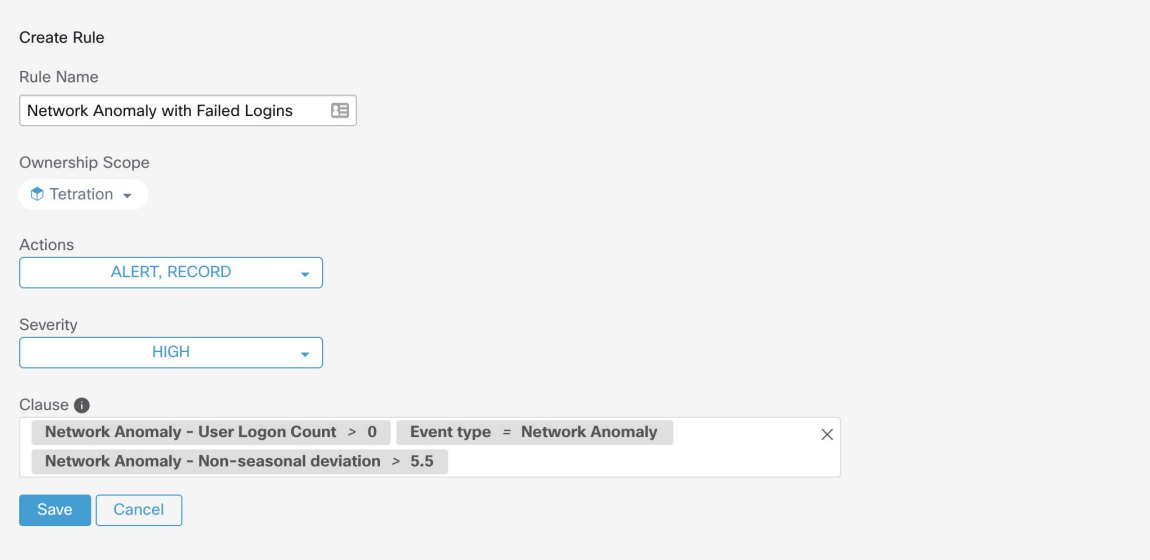
以下は、データセンター用のルールを調整するためのネットワーク異常イベントの他の属性です。

属性	説明
ホスト名	このイベントを発行するワークロードのホスト名。
タイムスタンプ（エポックミリ秒単位）	イベントのタイムスタンプ（ミリ秒単位）。
PCR 偏差	過去の複数の標準偏差としてのイベント時間の平均に基づく PCR の偏差。
非季節的偏差	これは、季節性パターンを（cron ジョブなどによって）削除した後の PCR 偏差です。非季節的偏差の値は常に 6.0 以上です。
PCR	生産者コンシューマ比率（Producer Consumer Ratio）。

属性	説明
EIR	送受信比（Egress Ingress Ratio）は、Egress App Byte Count の合計と Ingress App Byte Count の比率です。
送信側アプリバイト数	ワークロードから流出するパケットコンテンツ（ヘッダーを除く）の合計バイト数である、送信側アプリケーションのバイト数。
受信側アプリバイト数	ワークロードに流入するパケットコンテンツ（ヘッダーを除く）の合計バイト数である、受信側アプリケーションのバイト数。
プロトコル	PCR 時系列が計算されるプロトコル。現在、サポートされているプロトコルはTCP、UDP、および Aggregate です。Aggregate PCR は、TCP、UDP、およびICMPバイト数の合計に基づいて計算されます。
ユーザーログオン数	過去約15分間のワークロードでのユーザーログオンイベントの数。一致するルールがあるかどうかを考慮に入れない、ユーザーログオンイベントの数です。ユーザーログオンイベントの詳細を知るには、関心のあるワークロードのイベントを記録し、[フォレンジック分析（Forensics Analysis）] ページでそれらを表示するルールを定義する必要があります。
ユーザーログオン失敗数	過去約15分間のワークロードでのユーザーログオン失敗イベントの数。一致するルールがあるかどうかを考慮に入れない、ユーザーログオン失敗イベントの数です。ユーザーログオン失敗イベントの詳細を知るには、関心のあるワークロードのイベントを記録し、[フォレンジック分析（Forensics Analysis）] ページでそれらを表示するルールを定義する必要があります。
未確認コマンド数	過去約15分間のワークロードでの未確認コマンドイベントの数。一致するルールがあるかどうかを考慮に入れない、未確認コマンドイベントの数です。未確認コマンドイベントの詳細を知るには、関心のあるワークロードのイベントを記録し、[フォレンジック分析（Forensics Analysis）] ページでそれらを表示するルールを定義する必要があります。

属性	説明
日時（UTC）- 年	イベント時間の年。
日時（UTC）- 月	イベント時間の月（1、2、...。
日時（UTC）- 日	イベント時間の日（1、2、...。
日時（UTC）- 時間	イベント時間の時間（1、2、...、24）。
日時（UTC）- 分	イベント時間の分（1、2、...、60）。
日時（UTC）- 秒	イベント時間の秒（1、2、...、60）。
日時（UTC）- 曜日	イベント時間の曜日（0～7、月曜日から日曜日に対応）。

図 *371*：ネットワーク異常イベントのフォレンジックルールの定義



いくつかのサンプルルールを以下に示します。

リスト 7.10.1.1.1：UDP のみのネットワーク異常を検出します。

```
Event Type = Network Anomaly AND Network Anomaly Is = Protocol - UDP
```

リスト 7.10.1.1.2：名前に *sensitiveDataServer* が含まれているワークロードのサブセットの送信側アプリにおけるバイト数にしきい値を設定して、季節的なパターンを削除すると、非常に大きな偏差が検出されます（検出された場合）。

```
Event Type = Network Anomaly AND Network Anomaly - Non-seasonal Deviation > 10.0)
AND Network Anomaly - Egress App Byte Count > 1000000
AND Network Anomaly - Host Name CONTAINS sensitiveDataServer
```

リスト7.10.1.1.3：未確認コマンドイベントを伴うワークロードのネットワーク異常イベントが検出されます（ネットワーク異常イベントが毎日7.30AM UTC ～ 7.35AM UTCに発生することを除く）。

```
    Event Type = Network Anomaly AND Network Anomaly - Unseen Command Count > 0
    AND ( Network Anomaly - Date Time (UTC) - Hour != 7
    OR Network Anomaly - Date Time (UTC) - Minute < 30 OR Network Anomaly - Date Time
(UTC) - Minute > 35 )
```

ルール アクション

アクション	説明
RECORD	一致するイベントは、ネットワーク異常スコアに影響を与えます。該当するイベントをセキュリティダッシュボードまたは[ネットワーク異常（Network Anomalies）]タブで見つけることができます。
ALERT	一致するイベントは、現在のアラートと選択したアラートのパブリッシャの選択に表示されます。

次のセクションでは、UI で検出されたネットワーク異常イベントが表示される場所について詳しく説明します。

ネットワーク異常イベントが表示される場所

（注） ネットワーク異常イベントは、現在、[フォレンジック分析（Forensics Analysis）]ページには表示されていません。ネットワーク異常イベントは、次のページで確認できます。

- **セキュリティダッシュボード**：**RECORD** アクションを持つルールに一致するネットワーク異常イベントは、セキュリティダッシュボードの [ネットワーク異常スコア（Network Anomaly Score）] セクションで見つけることができます。スコアが最高ではない（100 未満の）ワークロードがある場合、ワークロード名をクリックすると、そのワークロードのPCR時系列とネットワーク異常イベントを表示できます。ネットワーク異常イベントテーブルの各行の右端には、対応するネットワーク異常イベントの前後におけるフローやその他のフォレンジックイベントを検索するのに役立つアクションリンクが表示されます。ネットワーク異常スコアレポートの既知の遅延については、「ネットワーク異常の遅延」を参照してください。

図 *372* : セキュリティダッシュボードのネットワーク異常スコア

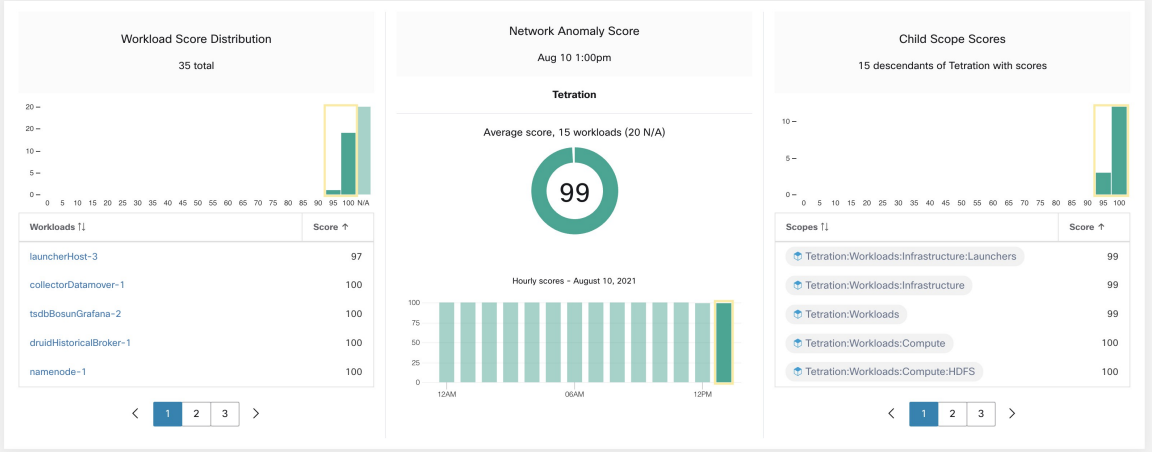
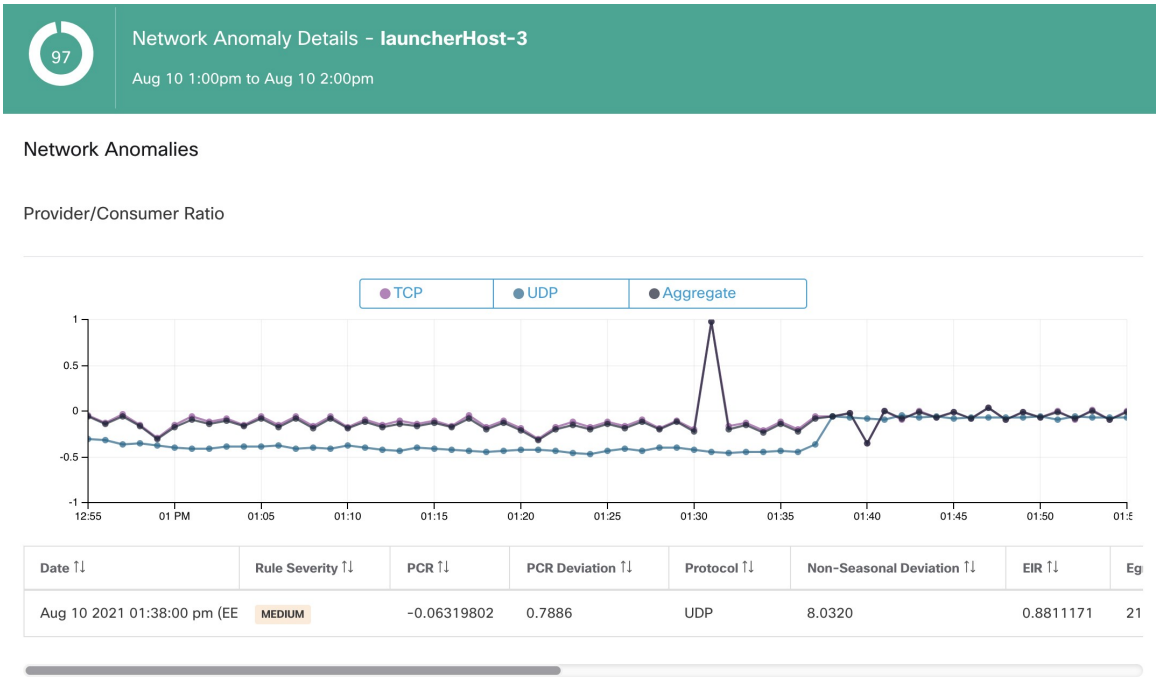
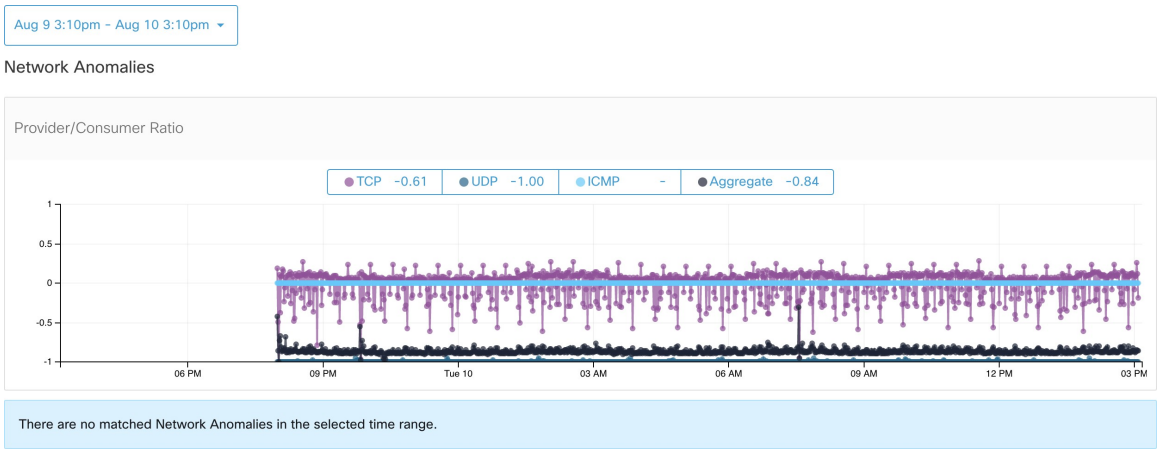


図 *373* : セキュリティダッシュボードのワークロード別にドリルダウンされたネットワーク異常スコア



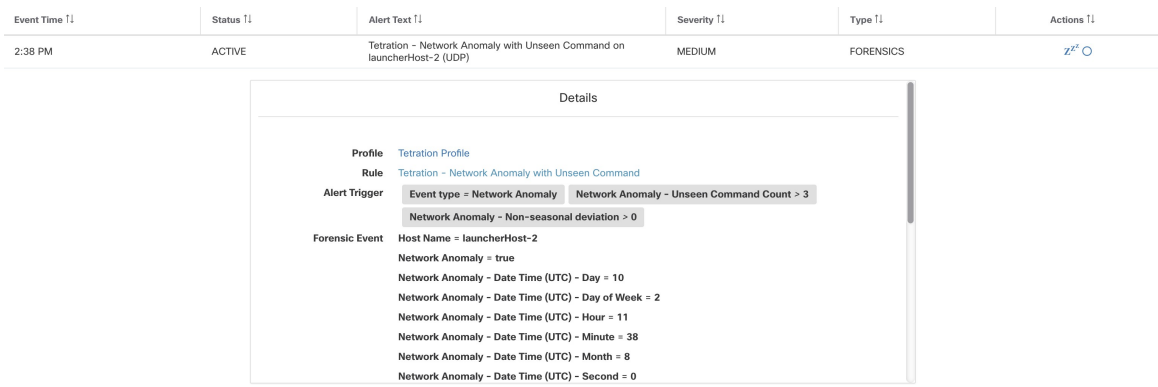
- [ネットワーク異常（Network Anomalies）]タブ：このページでは、PCR時系列グラフと、**RECORD**アクションを持つルールに一致するネットワーク異常イベントを確認できます。このページに表示される内容は、セキュリティダッシュボードでワークロード名をクリックして表示される内容と非常によく似ています。

図 *374 : [ワークロードプロファイル（Workload Profile）] ページの [ネットワーク異常（Network Anomaly）] タブ*



- **アラート**：ネットワーク異常ルールが **ALERT** アクションを指定して設定されている場合、一致したイベントは [アラート（Alerts）] ページに表示され、アラートパブリッシャからも利用できます。現在のアラート （798 ページ）

図 *375 :* ネットワーク異常アラート



ルールの重大度とネットワーク異常スコア

ネットワーク異常スコアは、フォレンジックスコアと同様に計算されます。ネットワーク異常スコアはワークロードごとに計算されます。ワークロードのネットワーク異常スコアは、該当範囲で有効になっているプロファイルに基づいて、そのワークロードで観察されたネットワーク異常イベントから得られます。スコアが 100 の場合、有効なプロファイルで設定されたルール経由でネットワーク異常イベントが観察されなかったことを意味します。スコアが 0 の場合、即時のアクションが必要なネットワーク異常イベントが検出されたことを意味します。

- 重大度が REQUIRES IMMEDIATE ACTION のネットワーク異常イベントは、範囲全体のスコアを 0 に減らします。
- 重大度が CRITICAL のネットワーク異常イベントは、ワークロードの影響スコアを 10 減らします。

- 重大度が HIGH のネットワーク異常イベントは、ワークロードの影響スコアを 5 減らします。
- 重大度が MEDIUM のネットワーク異常イベントは、ワークロードの影響スコアを 3 減らします。
- 重大度が LOW のネットワーク異常イベントは、ネットワーク異常スコアに影響を与えません。これは、信号の品質がまだ調整中であり、ノイズが多い可能性がある新しいルールに推奨されます。

ワークロードごとに、合計影響スコアが 5 分ごとに集計され、その 5 分以内の該当するワークロードのスコアが計算されます。

ネットワーク異常機能が有効になっているセンサータイプがないワークロードの場合、ネットワーク異常のスコアは N/A となります。

PCR データとネットワーク異常イベントの保持

PCR データとネットワーク異常イベントは 7 日間保持されます。

ネットワーク異常の遅延

セキュリティダッシュボードで報告されるネットワーク異常スコアには、5 分の遅延があります。たとえば、午前 10 時から午前 10 時 59 分までのワークロードのスコアは、午前 9 時 55 分から午前 10 時 54 分に発生するネットワーク異常イベントに基づいています。

警告

- 古いデータリークイベントは、ネットワーク異常イベントではなくデータリークイベントとして残ります。
- プロトコルごとのネットワーク異常検出は 3.3 の新機能であり、古いデータリークイベントではプロトコルは設定されません。

プロセスハッシュの異常検出

名前が示すように、この機能は、システム全体のプロセスバイナリハッシュの一貫性を評価することで、プロセスハッシュの異常を検出します。この機能には次のような開発背景があります。同じセットアップ構成から複製された Apache Web サーバーのファームがあるとします（これらのサーバーが同じ自動化スクリプトから展開されるなど）。その場合、すべてのサーバー上の httpd バイナリのハッシュは同じであると予想されます。不一致がある場合は異常であり、さらに調査が必要であると考えられます

正式には、同じルート範囲内のワークロード全体で、実行可能ファイルのバイナリパス、OS バージョン、パッケージ情報（該当する場合）1 の組み合わせが同一である一連のプロセスとしてプロセスグループを定義します。



（注） パッケージ情報は 3.4 リリース以降で含まれます。それ以前のリリースでは、プロセスグループは、実行可能ファイルのバイナリパスと OS バージョンの組み合わせのみに基づいて定義されていました。

上記の例では、すべての Apache Web サーバーが CentOS 7.7 および同じルート範囲で httpd 2.4.43 を実行していると仮定した場合、対応するプロセスグループは（すべてのサーバーにわたって）同じ組み合わせを持つプロセスのセットとなります（バイナリパスが `/usr/sbin/ httpd` で、OS バージョンが `CentOS-7.7` で、パッケージバージョンが `httpd-2.4.43`）。同じプロセスグループ内のすべてのバイナリのハッシュは同一であると予想されるため、不一致が検出されると異常が表示されます。

この機能は、異常なプロセスハッシュを検出するだけでなく、ユーザーがユーザーがアップロードしたファイルハッシュしたフラグ付きリストに表示されるプロセスハッシュも検出します。既知のマルウェアハッシュのリストがあり、それらのハッシュに関連付けられたプロセスが実行されているかどうかを知りたいということが開発の背景にあります。

誤報を減らすために、NIST が提供する National Software Reference Library の Reference Data Set（RDS）（NIST RDS データセットとも呼ばれます）を良性リストとして使用します。良性ハッシュは「安全」と見なされます（NIST RDS データセットを有効にする方法については、脅威インテリジェンス（931 ページ）を参照してください）。独自の良性ハッシュリストをユーザーがアップロードしたファイルハッシュすることもできます。

NIST RDS データセットに加えて、**Cisco Secure Workload Hash Verdict** サービスもキュレートしています。このサービスを有効にすると、既知のマルウェアハッシュが発生した場合に、悪意のあるハッシュとして検出されます。一方、ハッシュが既知で正当な場合は、異常分析で良性とマークされることもあります。非常に大規模なデータセットと迅速な更新により、ワークロードで実行されているプロセスの承認または危険信号の付与に使用できるすべての既知の正当なプロセスハッシュをカバーしているため、Secure Workload Hash Verdict は Secure Workload クラウド経由でのみ利用できます。「自動更新」を参照して、アプライアンスから Secure Workload Hash Verdict サービスにアクセスできることを確認してください。

この機能では、**プロセスハッシュスコア**と呼ばれるセキュリティスコアが出力されます。このスコアは 1 時間ごとに計算され、出力されます。他のすべてのセキュリティスコアと同様に、プロセスハッシュスコアは高いほど優れています。プロセスハッシュに固有のポイント:

- ハッシュスコアが 0 の場合、ハッシュにフラグが設定されているか、または悪意のあることを意味します。
- ハッシュスコアが 100 の場合、ハッシュが良性であるか、ワークロード全体で一貫している（不一致がない）ことを意味します。
- ハッシュスコアが 1 から 99 までの場合、ハッシュが異常であると見なされることを意味します（つまり、いくつかの不一致があります）。

ワークロードのプロセスハッシュスコアは、そのワークロードで観察されたすべてのハッシュの最小プロセスハッシュスコアであり、0 はシステム内にフラグ付きまたは悪意のあるプロセスハッシュがあることを意味し、100 はシステムでハッシュ異常が観察されていないことを意味します。

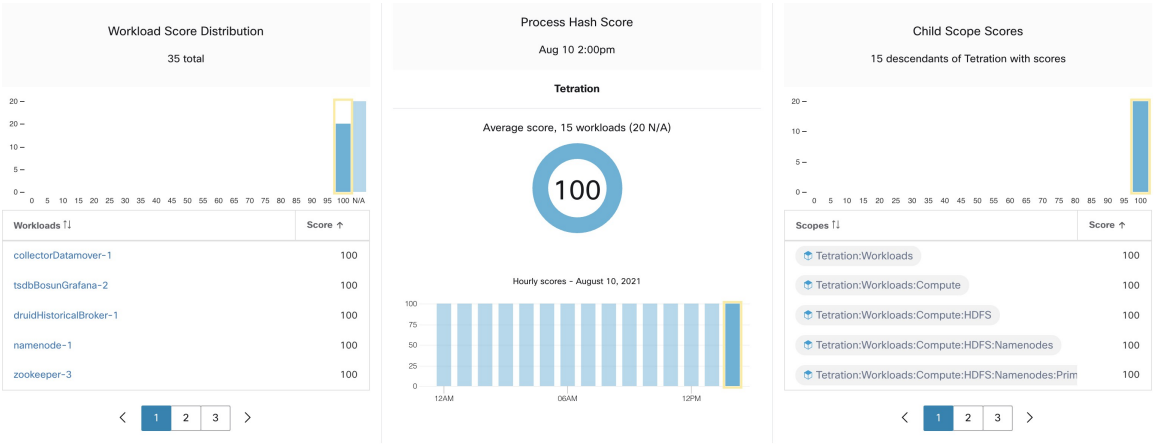
プロセスハッシュ機能を有効にする方法

プロセスハッシュ機能は、優れた可視性エージェントと適用エージェントでデフォルトで有効になっています。フォレンジック設定は必要ありません。システムにそのようなエージェントがある場合は、システムが起動してから 2 時間以内にスコア表示が開始されます。

プロセスハッシュスコアが表示される場所

- セキュリティダッシュボード：

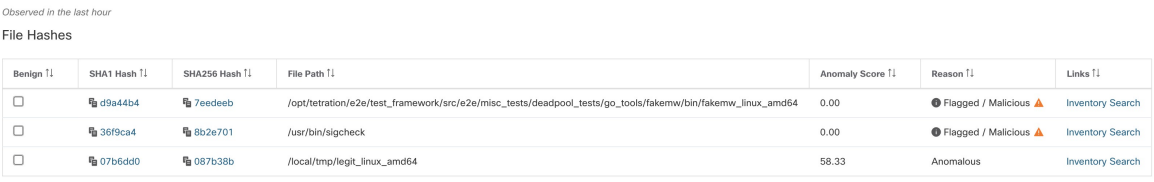
図 *376 :* セキュリティダッシュボードの *[*プロセスハッシュスコア（*Process Hash Score***）*]* セクション**



セキュリティダッシュボードの [プロセスハッシュスコア（Process Hash Score）] セクションセキュリティ ダッシュボード （947 ページ）

- [ワークロードプロファイル（Workload Profile）] ページ/[ファイルハッシュ（File Hashes）] タブ：

図 *377 : [*ワークロードプロファイル（*Workload Profile***）*]* ページの *[*ファイルハッシュ（***File Hashes***）*]* タブ**



[ワークロードプロファイル（Workload Profile）] ページの [ファイルハッシュ（File Hashes）] タブワークロード プロファイル （462 ページ）

プロセスハッシュスコアの計算方法

各プロセスハッシュについて、次のようにスコアを計算します。

1. ハッシュにフラグが付けられているか、悪意がある場合：`score = 0`
2. 上記とは異なり、ハッシュがクリーンな場合：`score = 100`
3. 上記とは異なり、ハッシュに何らかの異常がある場合：`score` が `[1, 99]` の範囲内。高いほど良い
4. 上記とは異なる場合：`score = 100`

（3）のスコアを計算するロジックは、まずハッシュのマイノリティスコア（同じルート範囲に属するワークロードの母集団における該当するハッシュの母集団の比率を1から引いた値）を計算し、次にそれを範囲 `[0.0, 1.0]` にマッピングするというものです。ハッシュのマイノリティスコアが 0.5 を超える場合、情報関数 `-log2(x)` を使用して、スコアを範囲 `[1.0, 99.0]` に再度マッピングします。上記の Apache Web サーバーファームの例を取り上げ、`httpd` のハッシュについて考えてみましょう。以下にいくつかのシナリオを示します。

- `httpd` がファーム内の 1000 台のサーバーに 2 つのハッシュ値（`h1` と `h2`）を持っているとします。1 台のサーバーで `h1`、残りの 999 台のサーバーでは `h2` です。この場合、次のように計算します。
  - `population_ratio(h1) = 0.001, population_ratio(h2) = 0.999`. 実行されるアクション
  - `minority_score(h1) = 0.999, minority_score(h2) = 0.001`. 実行されるアクション
  - `score(h1) = -log2(0.999) * 98 + 1 = 1.14;`
  - `minority_score(h2) < 0.5, h2` は異常とは見なされないため、`score(h2) = 100` となります。
- `httpd` がファーム内の 10 台のサーバーに 2 つのハッシュ値（`h1` と `h2`）を持っているとします。1 台のサーバーで `h1`、残りの 9 台のサーバーでは `h2` です。この場合、次のように計算します。
  - `population_ratio(h1) = 0.1, population_ratio(h2) = 0.9`. 実行されるアクション
  - `minority_score(h1) = 0.9, minority_score(h2) = 0.1`. 実行されるアクション
  - `score(h1) = -log2(0.9) * 98 + 1 = 15.90;`
  - `minority_score(h2) < 0.5, h2` は異常とは見なされないため、`score(h2) = 100` となります。
- `httpd` がファーム内の 2 台のサーバーに 2 つのハッシュ値（`h1` と `h2`）を持っているとします。一方のサーバーで `h1`、他方のサーバーでは `h2` です。この場合、次のように計算します。
  - `population_ratio(h1) = population_ratio(h2) = 0.5`. 実行されるアクション

- `minority_score(h1) = minority_score(h2) = 0.5`. 実行されるアクション
- `score(h1) = score(h2) = -log2(0.5) * 98 + 1 = 99.0`. これは、異常と見なされるハッシュに付けられる最高のスコアです。

- `httpd` がすべてのサーバーに 1 つのハッシュ値（`h1`）のみを持っているとします。この場合、`minority_score(h1) = 0.0 < 0.5` となります。したがって、異常とは見なされず、`score(h1) = 100` となります。

最終的に、ワークロードのプロセスハッシュスコアは、そのワークロードで観察されたすべてのハッシュの最小プロセスハッシュスコアになります。

`-log2(x)` 情報関数に関する追加情報は、こちらで見つけることができます。

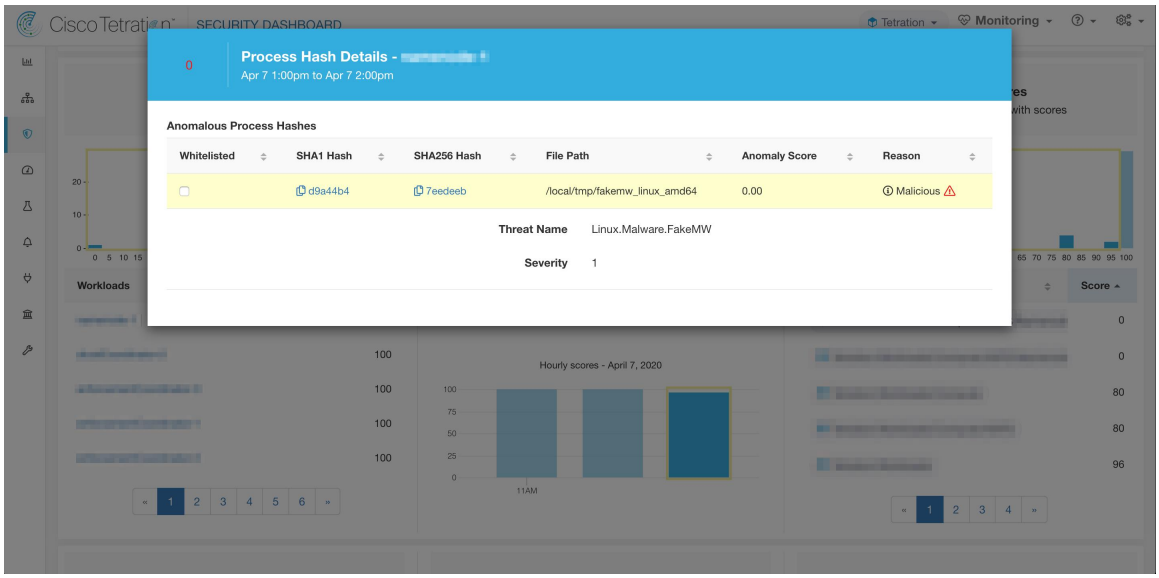
プロセスハッシュスコアの改善方法

ワークロードでのプロセスハッシュスコア 0 は、フラグ付きのプロセスハッシュまたは悪意のあるプロセスハッシュがそのワークロードに現れたことを意味します。このようなプロセスが再度実行されないようにすると、スコアが改善します。100 未満の正のプロセスハッシュスコアは、システム全体にプロセスハッシュの異常があることを意味します。悪意がある場合とそうでない場合がありますが、さらに調査が必要です。慎重に調査した結果、ハッシュが安全であると結論付けられた場合、それを良性リストに追加すると、スコアも改善します。ユーザーは [ファイルハッシュ（File Hashes）]/[プロセスハッシュの詳細（Process Hash Details）] ページで [良性（benign）] のチェックボックスをオンにするか、ユーザーがアップロードしたファイルハッシュすることで、異常なハッシュを「良性」として設定できます。

脅威情報の詳細

前述のように、Secure Workload Hash Verdict サービスが有効になっている場合、既知のマルウェアハッシュが発生すると、悪意があるとフラグが付けされます。その場合、悪意のあるハッシュの追加の脅威情報（脅威インテリジェンスプラットフォームを介して収集される）が提供されます。現在、追加の脅威データには、脅威の名前と重大度が含まれています。脅威名は脅威の名前で、重大度は脅威の重大度を示す 1 から 5 の値です。1 は重大度が最も低く、5 は重大度が最も高いことを意味します。

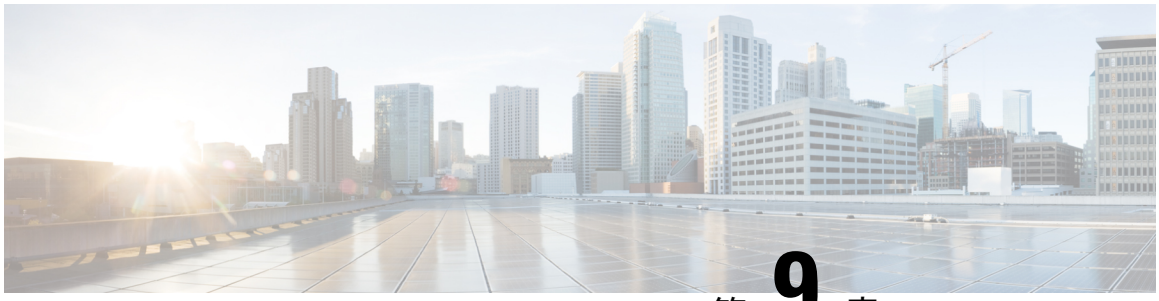
図 *378* : ユーザーは悪意のあるハッシュの行をクリックして、その脅威情報の詳細を表示できます



警告

- プロセスハッシュ分析タスクは1時間ごとに実行されますが、アクションによっては、予想されるスコアや結果がセキュリティダッシュボードに表示されるまでに最大2時間かかる場合があります。以下に例を示します。
  - ハッシュフラグ付きリストをアップロードし、そのリストにプロセスハッシュが表示された場合、スコアがセキュリティダッシュボードに反映されるまでに最大1時間かかる場合があります。
  - フラグ付きリストからハッシュを削除した場合、セキュリティダッシュボードでハッシュが完全に消去される（およびスコアが反映される）までに最大2時間かかる場合があります。
- 保持：
  - プロセスハッシュ分析の詳細な結果は、少なくとも 7 日間保持されます。
- [ワークロードプロファイル（Workload Profile）]ページの[ファイルハッシュ（File Hashes）]タブには、過去 1 時間に分析されたプロセスハッシュの詳細のみが表示されます。
- 以前のバージョンの優れた可視化エージェントと適用エージェント、および AnyConnect エンドポイントは、SHA256 ハッシュ値のみを報告します。したがって、SHA1 ハッシュのフラグ付きリストや良性リストとの照合は、これらのエージェントではサポートされていません。
- プロセスハッシュスコアは、特定のルート範囲について計算されます。ワークロードが複数のルート範囲に属している場合、そのワークロードが属するすべてのルート範囲の最小スコアがそのワークロードのプロセスハッシュスコアになります。

- プロセスハッシュの異常分析での誤認アラームをさらに減らすために、すべての Secure Workloadエージェントのバイナリをそれらのファイルパスに従って良性としてマークします。これらのハッシュがユーザー定義のハッシュリストに表示されない場合、またはSecure Workload Hash Verdict サービスによってフラグ付けされない場合にのみこのメカニズムが発生します。



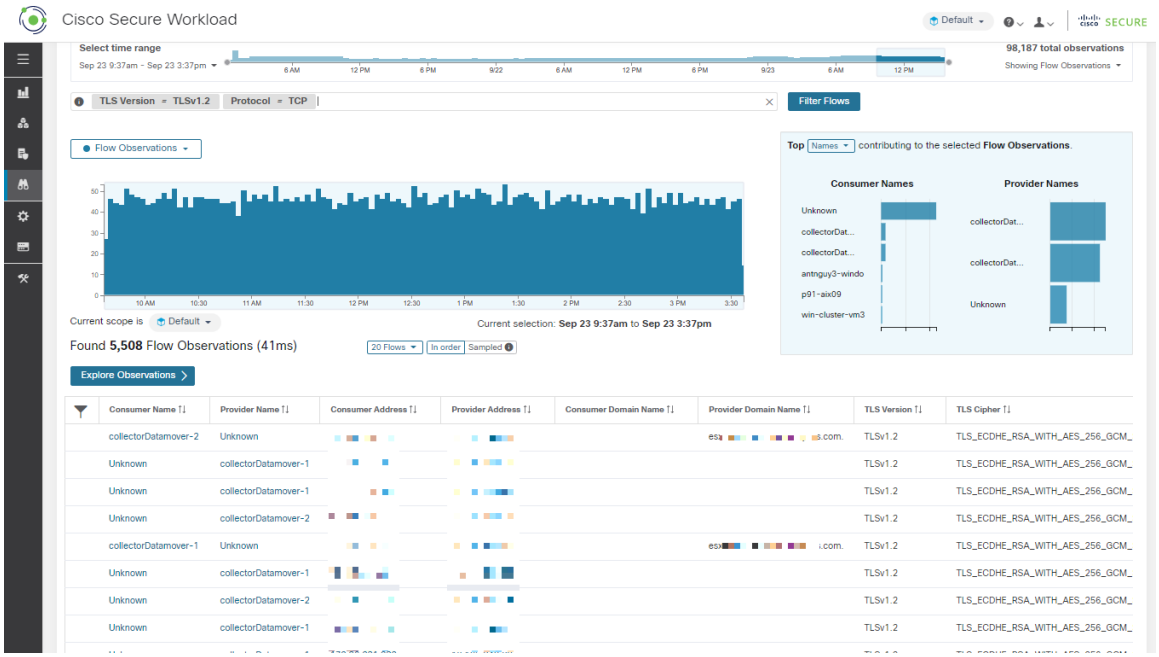
第 9 章

フロー

左側のナビゲーションメニューの**[調査（Investigate）]**>**[トラフィック（Traffic）]**オプションから[フローの検索（Flow Search）]ページに移動します。このページでは、フローコーパスをすばやくフィルタリングおよびドリルダウンできます。基本単位となる「フロー観測データ」は、固有のフローごとに分単位で集計されます。フローには「コンシューマ」側と「プロバイダー」側の 2つがあり、コンシューマ側でフローが開始され、プロバイダーはコンシューマに応答します（例：それぞれ「クライアント」と「サーバー」になります）。それぞれの観測では、フローの各方向のパケット数、バイト数、およびその他のメトリックが分間隔で追跡されます。フローを迅速にフィルタリングできるだけでなく、[観測結果の確認（Explore Observations）]ボタンを使用してフローを可視化できます。フロー観測結果のリストをクリックして、そのフローのライフタイム全体を通した遅延、パケット数、バイト数などのフローの詳細情報を表示できます。

警告 優れた可視化エージェントや適用エージェントを備えたホストの場合、Secure Workload はフローデータをそのフローを提供または利用するプロセスと関連付けることができます。その結果、プロセスの起動に使用される**データベースやAPIのクレデンシャルといった機密情報**を含む可能性のあるすべてのコマンドライン引数を分析や表示に利用できます。

図 *379* : フローの概要



- コーパスセレクタ （758 ページ）
- 列とフィルタ （759 ページ）
- フィルタ処理された時系列 （765 ページ）
- 上位 N 件チャート （767 ページ）
- 観測リスト （768 ページ）
- 観測結果の確認 （770 ページ）
- クライアントサーバーの分類 （772 ページ）
- カンバセーションモード （776 ページ）

コーパスセレクタ

図 *380* : コーパスセレクタ

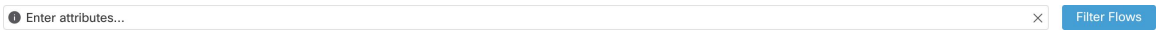


これは、コーパス全体における現在の [範囲（Scope）] の、フィルタ処理されていない要約された時系列データです。このコンポーネントの目的は、表示されている日付範囲を把握し、コンポーネント内をドラッグしてその日付範囲を簡単に変更できるようにすることです。チャート内のデータは、選択すべき時間範囲の決定に役立つ場合があります。表示するメトリックの種類は選択できます。デフォルトでは、[フロー観測（flow observations）] の数が表示されます。

コーパスセレクタは、現在およそ [20億のフロー観測（2 billion flow observations）] を選択できるようになっています。

列とフィルタ

図 *381* : フィルタ入力



ここで、検索結果を絞り込むためのフィルタを定義します。[フィルタ（Filters）] という単語の横にある [(?)] アイコンをクリックすると、すべての次元が表示されます。ユーザーラベルデータについては、これらの列も適切な間隔で使用できます。この入力は and、or、not、および括弧などのキーワードもサポートしており、これらを使用してより複雑なフィルタを表現します。たとえば、IP 1.1.1.1 と 2.2.2.2 の間の指示に依存しないフィルタは次のように記述できます。

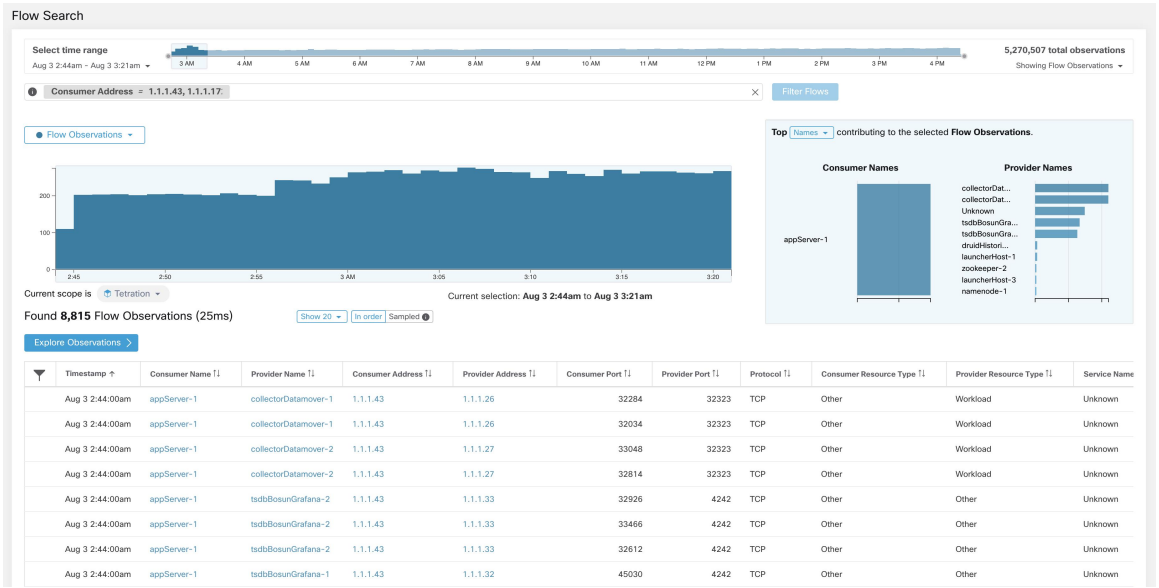
Consumer Address = 1.1.1.1 and Provider Address = 2.2.2.2 or Consumer Address = 2.2.2.2 and Provider Address = 1.1.1.1

And to additionally filter on Protocol = TCP:

Consumer Address = 1.1.1.1 and Provider Address = 2.2.2.2 or Consumer Address = 2.2.2.2 and Provider Address = 1.1.1.1

フィルタ入力機能は、「-」を範囲クエリに変換することで、ポート、コンシューマアドレス、プロバイダーアドレスの「,」と「-」もサポートします。以下に有効なフィルタの例を示します。

図 *382* : 例：フィルタ入力でコンシューマアドレスの「,」がサポートされている



Timestamp	Consumer Name	Provider Name	Consumer Address	Provider Address	Consumer Port	Provider Port	Protocol	Consumer Resource Type	Provider Resource Type	Service Name
Aug 3 2:44:00am	appServer-1	collectorDatamover-1	1.1.1.43	1.1.1.26	32284	32323	TCP	Other	Workload	Unknown
Aug 3 2:44:00am	appServer-1	collectorDatamover-1	1.1.1.43	1.1.1.26	32034	32323	TCP	Other	Workload	Unknown
Aug 3 2:44:00am	appServer-1	collectorDatamover-2	1.1.1.43	1.1.1.27	33048	32323	TCP	Other	Workload	Unknown
Aug 3 2:44:00am	appServer-1	collectorDatamover-2	1.1.1.43	1.1.1.27	32814	32323	TCP	Other	Workload	Unknown
Aug 3 2:44:00am	appServer-1	tsdbBosunGrafana-2	1.1.1.43	1.1.1.33	32926	4242	TCP	Other	Other	Unknown
Aug 3 2:44:00am	appServer-1	tsdbBosunGrafana-2	1.1.1.43	1.1.1.33	33466	4242	TCP	Other	Other	Unknown
Aug 3 2:44:00am	appServer-1	tsdbBosunGrafana-2	1.1.1.43	1.1.1.33	32612	4242	TCP	Other	Other	Unknown
Aug 3 2:44:00am	appServer-1	tsdbBosunGrafana-1	1.1.1.43	1.1.1.32	45030	4242	TCP	Other	Other	Unknown

図 *383 :* 例：フィルタ入力でコンシューマアドレスの範囲クエリがサポートされている

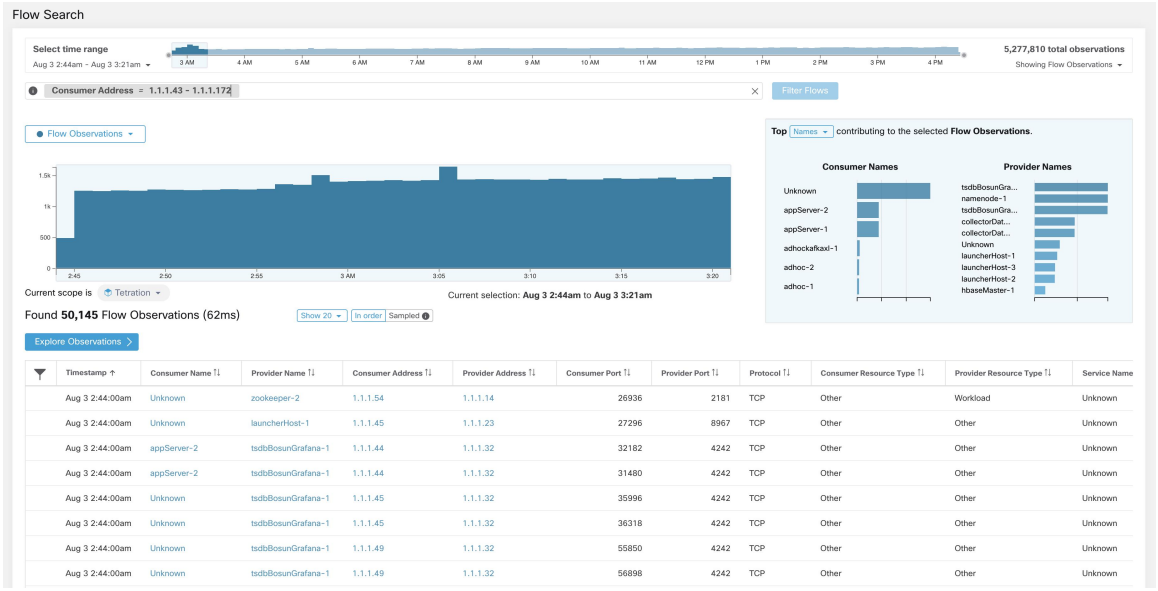


表 *33 :* 利用可能な列とフィルタ

列（**API** で公開される名前）	説明	ソース（**Source**）
[コンシューマアドレス（Consumer Address）]（src_address）	CIDR 表記を使用してサブネットまたは IP アドレスを入力します（例：10.11.12.0/24）。コンシューマアドレスが入力された IP アドレスまたはサブネットと重複するフロー観測を一致させます。	ソフトウェアエージェントと Ingest アプライアンス
[プロバイダーアドレス（Provider Address）]（dst_address）	CIDR 表記を使用してサブネットまたは IP アドレスを入力します（例：10.11.12.0/24)。プロバイダーアドレスが入力された IP アドレスまたはサブネットと重複するフロー観測を一致させます。	ソフトウェアエージェントと Ingest アプライアンス
[コンシューマドメイン名（Consumer Domain Name）]	（コンシューマ IP アドレス/サブネットに関連付けられている）コンシューマドメイン名が入力されたコンシューマドメイン名と重複するフロー観測を一致させます。	ソフトウェアエージェントと Anyconnect
[プロバイダードメイン名（Provider Domain Name）]	（プロバイダー IP アドレス/サブネットに関連付けられている）プロバイダードメイン名が入力されたプロバイダードメイン名と重複するフロー観測を一致させます。	ソフトウェアエージェントと Anyconnect

列（API で公開される名前）	説明	ソース（Source）
[コンシューマホスト名（Consumer Hostname）]（src_hostname）	コンシューマホスト名が入力されたホスト名と重複するフローを一致させます。	ソフトウェアエージェントと Anyconnect
[プロバイダーホスト名（Provider Hostname）（dst_hostname）	プロバイダーのホスト名が入力されたホスト名と重複するフローを一致させます。	ソフトウェアエージェントと Anyconnect
[コンシューマ適用グループ（Consumer Enforcement Group）]（src_enforcement_epg_name）	コンシューマ適用グループは、コンシューマに一致する適用ポリシー内のフィルタ（範囲、インベントリフィルタ、またはクラスタ）の名前です。	内線
[プロバイダー適用グループ（Provider Enforcement Group）]（dst_enforcement_epg_name）	プロバイダー適用グループは、プロバイダーに一致する適用ポリシー内のフィルタ（範囲、インベントリフィルタ、またはクラスタ）の名前です。	内線
[コンシューマ分析グループ（Consumer Analysis Group）]	コンシューマ分析グループは、コンシューマに一致する分析済みポリシー内のフィルタ（範囲、インベントリフィルタ、またはクラスタ）の名前です。	内線
[プロバイダー分析グループ（Provider Analysis Group）]	プロバイダー分析グループは、プロバイダーに一致する分析済みポリシー内のフィルタ（範囲、インベントリフィルタ、またはクラスタ）の名前です。	内線
[コンシューマ範囲（Consumer Scope）（src_scope_name）	指定された範囲にコンシューマが属するフローを一致させます。	内線
[プロバイダー範囲（Provider Scope）]（dst_scope_name）	指定された範囲にプロバイダーが属するフローを一致させます。	内線
[コンシューマポート（Consumer Port）]（src_port）	コンシューマポートが入力されたポートと重複するフローを一致させます。	ソフトウェアエージェント、ERSPAN および NetFlow
[プロバイダーポート（Provider Port）]（dst_port）	プロバイダーポートが入力されたポートと重複するフローを一致させます。	ソフトウェアエージェント、ERSPAN および NetFlow

列（API で公開される名前）	説明	ソース（Source）
[コンシューマの国（Consumer Country）]（src_country）	コンシューマの国が入力された国と重複するフローを一致させます。	内線
[プロバイダーの国（Provider Country）]（dst_country）	プロバイダーの国が入力された国と重複するフローを一致させます。	内線
[コンシューマの下位区分（Consumer Subdivision）]（src_subdivision）	コンシューマの下位区分が入力された下位区分（都道府県）と重複するフローを一致させます。	内線
[プロバイダーの下位区分（Provider Subdivision）]（dst_subdivision）	プロバイダーの下位区分が入力された下位区分（都道府県）と重複するフローを一致させます。	内線
[コンシューマ自律システム構成（Consumer Autonomous System Organization）]（src_autonomous_system_organization）	コンシューマ自律システム構成が入力された自律システム構成（ASO）と重複するフローを一致させます。	内線
[プロバイダー自律システム構成（Provider Autonomous System Organization）]（dst_autonomous_system_organization）	プロバイダー自律システム構成が入力された自律システム構成（ASO）と重複するフローを一致させます。	内線
[プロトコル（Protocol）]（proto）	フロー観測をプロトコルタイプ（TCP、UDP、ICMP）でフィルタ処理します。	ソフトウェアエージェントと Ingest アプライアンス
[アドレスタイプ（Address Type）]（key_type）	フロー観測をアドレスタイプ（IPv4、IPv6、DHCPv4）でフィルタ処理します。	ソフトウェアエージェントと Ingest アプライアンス
[順方向TCPフラグ（Fwd TCP Flags）]	フロー観測をフラグ（SYN、ACK、ECHO）でフィルタ処理します。	ソフトウェアエージェント、ERSPAN および NetFlow

列（API で公開される名前）	説明	ソース（Source）
[逆方向TCPフラグ（Rev TCP Flags）]	フロー観測をフラグ（SYN、ACK、ECHO）でフィルタ処理します。	ソフトウェアエージェント、ERSPAN および NetFlow
[順方向プロセスUID（Fwd Process UID）]（fwd_process_owner）	フロー観測をプロセス所有者 UID（root、admin、yarn、mapred）でフィルタ処理します。	ソフトウェアエージェント
[逆方向プロセスUID（Rev Process UID）]（rev_process_owner）	フロー観測をプロセス所有者 UID（root、admin、yarn、mapred）でフィルタ処理します。	ソフトウェアエージェント
[順方向プロセス（Fwd Process）]（fwd_process_string）	フロー観測をプロセス（java、hadoop、nginx）でフィルタ処理します。「プロセス文字列の可視性の警告」を参照してください。	ソフトウェアエージェント
[逆方向プロセス（Rev Process）]（rev_process_string）	フロー観測をプロセス（java、hadoop、nginx）でフィルタ処理します。「プロセス文字列の可視性の警告」を参照してください。	ソフトウェアエージェント
[収集ルールのコンシューマ（Consumer In Collection Rules?）]	内部コンシューマのみを一致させます。	内線
[収集ルールのプロバイダー（Provider In Collection Rules?）]	内部プロバイダのみを一致させます。	内線
[SRTT使用可能（SRTT Available）]	「true」または「false」の値を使用して、使用可能な SRTT 測定値を持つフローを一致させます（これは SRTT > 0 に相当します）。	内線
Bytes	バイトトラフィックバケットでフロー観測をフィルタ処理します。バイトトラフィックバケット値が=、<、>（2の累乗（0、2、64、1024）でバケット化）であるフローを一致させます。	ソフトウェアエージェントと Ingest アプライアンス

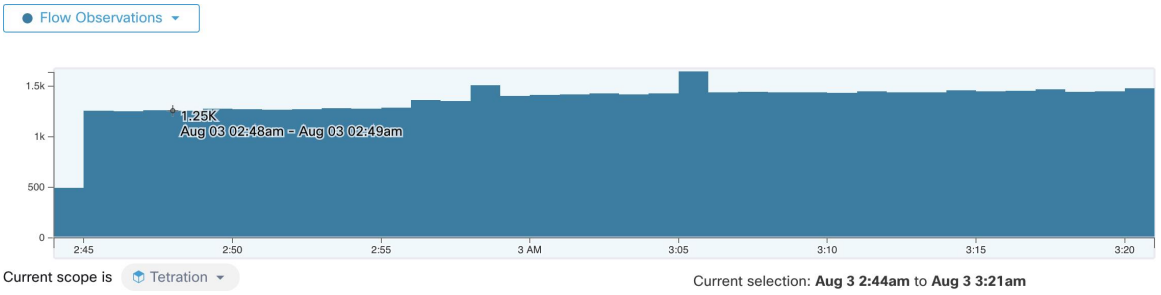
列（API で公開される名前）	説明	ソース（Source）
Packets	パケット トラフィック バケットでフロー観測をフィルタ処理します。パケット トラフィック バケット値が =、<、>（2 の累乗（0、2、64、1024）でバケット化）であるフローを一致させます。	ソフトウェア エージェントと Ingest アプライアンス
[フロー持続時間（マイクロ秒）（Flow Duration (µs)]	フロー持続時間バケットでフロー観測をフィルタ処理します。フロー持続時間バケット値が =、<、>（2 の累乗（0、2、64、1024）でバケット化）であるフローを一致させます。	内線
[データ持続時間（マイクロ秒）（Data Duration (µs)）]	データ持続時間バケットでフロー観測をフィルタ処理します。データ持続時間バケット値が =、<、>（2 の累乗（0、2、64、1024）でバケット化）であるフローを一致させます。	内線
[SRTT（マイクロ秒）（SRTT (µs)）]（srtt_dim_usec）	SRTT バケットでフロー観測をフィルタ処理します。SRTT バケット値が =、<、>（2 の累乗（0、2、64、1024）でバケット化）であるフローを一致させます。	ソフトウェア エージェント
[順方向パケットの再送信（Fwd Packet Retransmissions）]（fwd_tcp_pkts_retransmitted）	パケット再送信バケットでフロー観測をフィルタ処理します。パケット再送信バケット値が =、<、>（2 の累乗（0、2、64、1024）でバケット化）であるフローを一致させます。	ソフトウェア エージェント
[逆方向パケットの再送信（Rev Packet Retransmissions）]（rev_tcp_pkts_retransmitted）	パケット再送信バケットでフロー観測をフィルタ処理します。パケット再送信バケット値が =、<、>（2 の累乗（0、2、64、1024）でバケット化）であるフローを一致させます。	ソフトウェア エージェント
[ユーザーラベル（User Labels）]（* または user_ プレフィックス）	手動でアップロードされたカスタムラベルに関連付けられたユーザー定義データ。UI ではプレフィックスが * で、OpenAPI では user_ です。	CMDB
TLS バージョン	フローで使用される SSL プロトコルバージョン。	ソフトウェア エージェント
[TLS暗号方式（TLS Cipher）]	フローで SSL プロトコルによって使用されるアルゴリズムのタイプ。	ソフトウェア エージェント

列（API で公開される名前）	説明	ソース（Source）
[コンシューマエージェントタイプ（Consumer Agent Type）]	コンシューマ エージェント タイプを指定します。	内線
[プロバイダーエージェントタイプ（Provider Agent Type）]	プロバイダー エージェント タイプを指定します。	内線
[コンシューマリソースタイプ（Consumer Resource Type）]	ソースからコンシューマへのリソースのフローを表します。ワークロード、ポッド、サービスなどがあります。	内線
[プロバイダーリソースタイプ（Provider Resource Type）]	プロバイダーからコンシューマへのリソースのフローを表します。ワークロード、ポッド、サービスなどがあります。	内線

（注） フローデータは取り込み時にのみユーザーラベルでラベル付けされるため、ユーザーラベルを有効化してもすぐには表示されません。フロー検索でラベルが表示されるまでに数分かかる場合があります。また、使用可能なユーザーラベルは、[コーパスセレクタ（Corpus Selector）] で選択した部分によって異なります。有効化されたラベルがさまざまな時点で変更されている可能性があるためです。

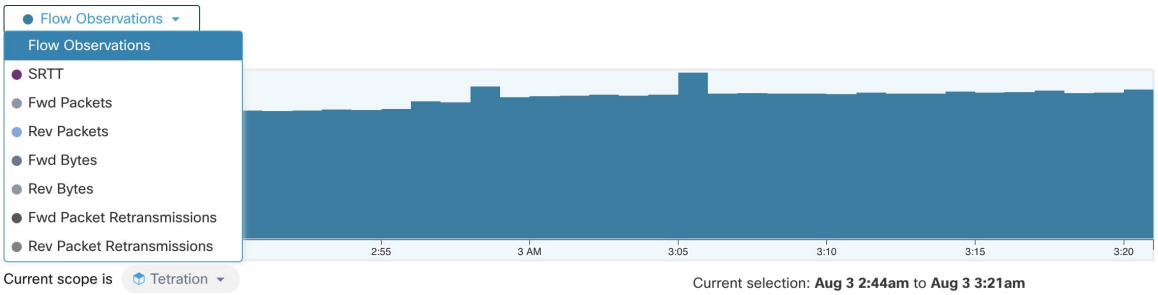
フィルタ処理された時系列

図 *384* : フィルタ処理された時系列



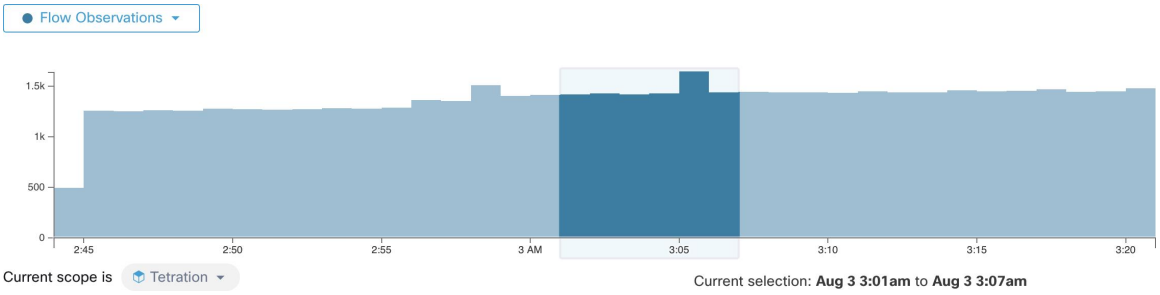
このコンポーネントは、選択した間隔（前述のコーパスセレクタ （758ページ）で行った選択）のさまざまなメトリックの集約合計を表示します。ドロップダウンを使用して、表示されるメトリックを変更します。

図 *385 : [*時系列（***Timeseries***）*]* ドロップダウン



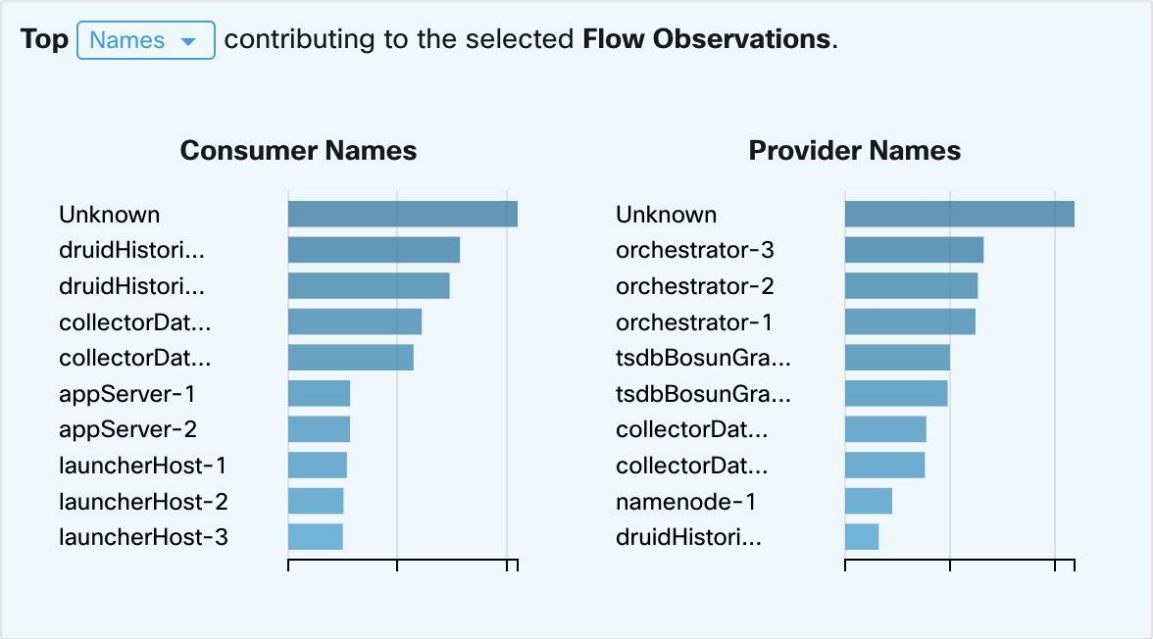
このコンポーネントでは、選択した間隔をさらに狭めることもできます。注目したいグラフの領域をクリックするだけで、上位Nチャートとその下のデータがすべて更新され、選択した間隔のデータのみが含まれます。

図 *386 :* 選択範囲による時系列チャート



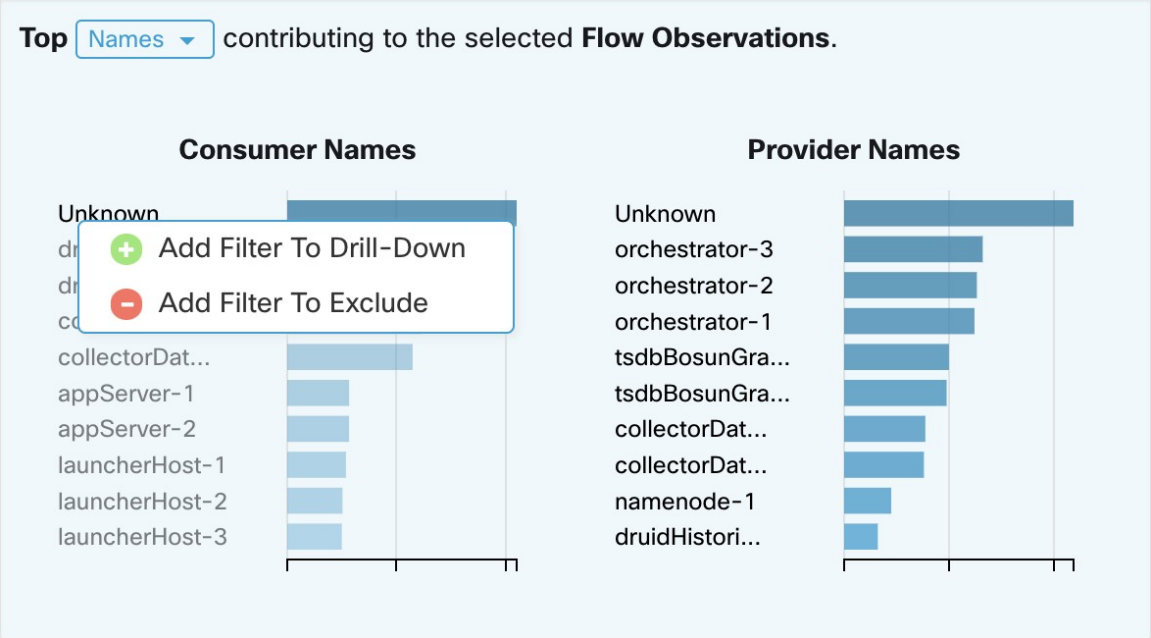
上位 N 件チャート

図 *387* : 上位 *N* 件チャート



このチャートには、左側のフィルタ済みの時系列チャートの選択に影響を与える上位N件の値が表示されます。時系列チャートのフロー観測でピークを選択し、上位N件チャートでホスト名を選択すると、それらのフロー観測に最も影響を与えるホスト名（コンシューマとプロバイダー）のリストが表示されます。また、時系列チャートがSRTTを表示するように設定されている場合、上位のホスト名には、選択したSRTTに最も影響を与えるものが表示されます。

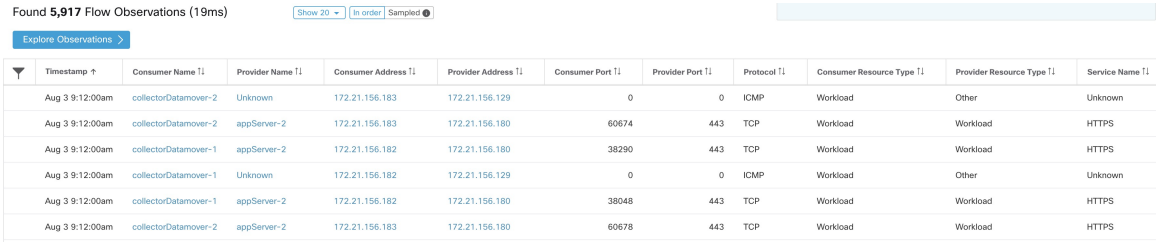
図 *388 :* ドリルダウンおよび除外



上位N件チャートのいずれかの項目をクリックすると、その値を「ドリルダウン」または「除外」できるメニューが表示されます。[ドリルダウン（Drill-down）] をクリックすると、結果をその値だけに限定するフィルタが追加されます。[除外（Exclude）]をクリックすると、結果からその値を除外するフィルタが追加されます。

（注） [ドリルダウン（Drill-down）] または [除外（Exclude）] をクリックした後、フィルタを有効にするには、[フィルタ（Filter）]ボタンを押す必要があります。これは、ページが途中で繰り返し更新されることなく、複数の「除外」アクションをすばやく実行できるようにするためです。

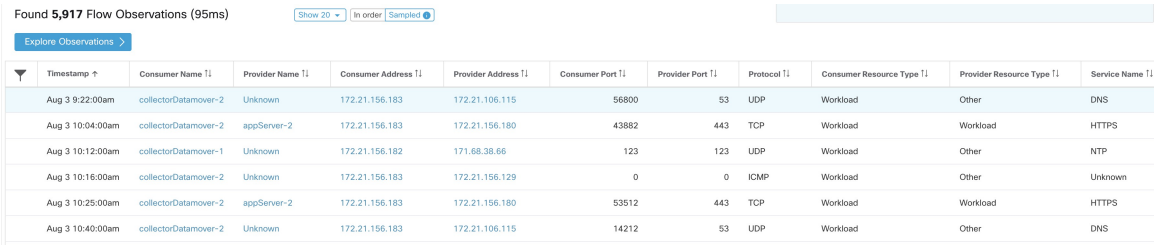
観測リスト



これは、上のページのフィルタおよび選択と一致する、実際の[フロー観測（Flow Observations）] のリストです。デフォルトでは、間隔の最初から 20 個がロードされます。ドロップダウンを使用すると、ロードされる数を増やせます。[順番（In Order）] ではなく [サンプル（Sampled）]

を使用して、選択された間隔からフロー観測のランダムセットの読み込みも可能です。[サンプル（Sampled）]設定は、間隔の最初から順番に読み込むのではなく、選択した間隔からより代表的なフロー観測のセットを取得するのに役立ちます。

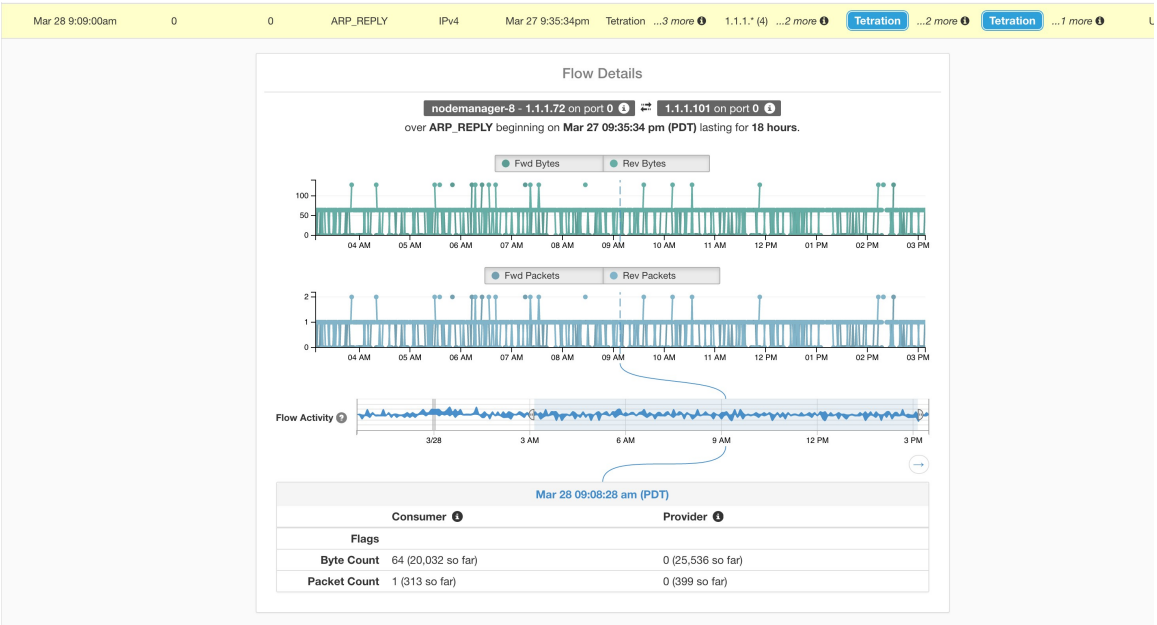
図 *389* : サンプル



フローの詳細

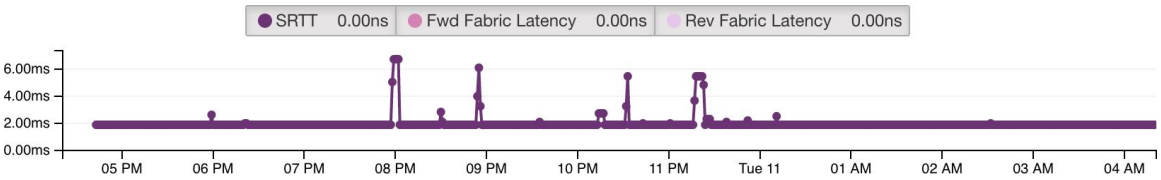
いずれかの行をクリックすると、その行の下にある [フローの詳細（Flow Details）] セクションが展開されます。フローの概要と、そのフローの存続期間中のさまざまなメトリックのグラフが表示されます。存続期間の長いフローの場合、サマリーチャートが下部に表示され、時系列データを表示するさまざまな間隔を選択できるようになります。

図 *390* : フローの詳細



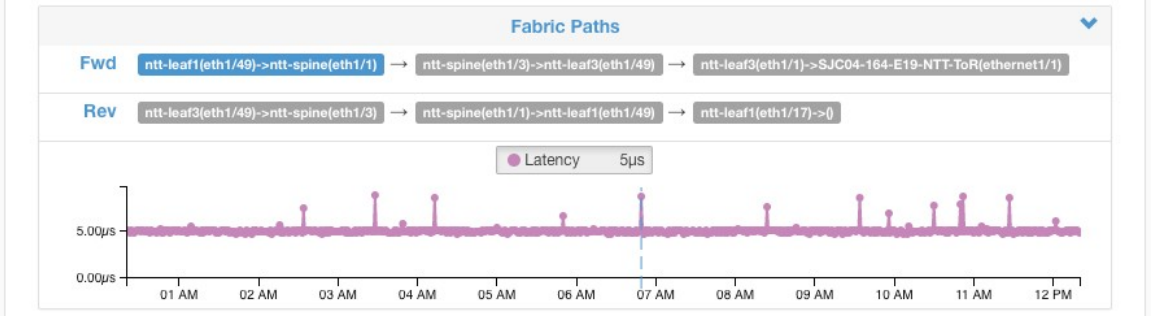
ファブリックパス情報でラベル付けされたフローの場合、[順方向/逆方向ファブリックレイテンシ（Fwd/Rev Fabric Latency）] と [SRTT] が使用可能になります。[順方向/逆方向バーストインジケータ（Fwd/Rev Burst Indicators）] および [順方向/逆方向バースト+ドロップインジケータ（Fwd/Rev Burst+Drop Indicators）] など、他のメトリックの時系列グラフが表示される場合があります。「互換性」を参照してください。

図 *391 :* 遅延



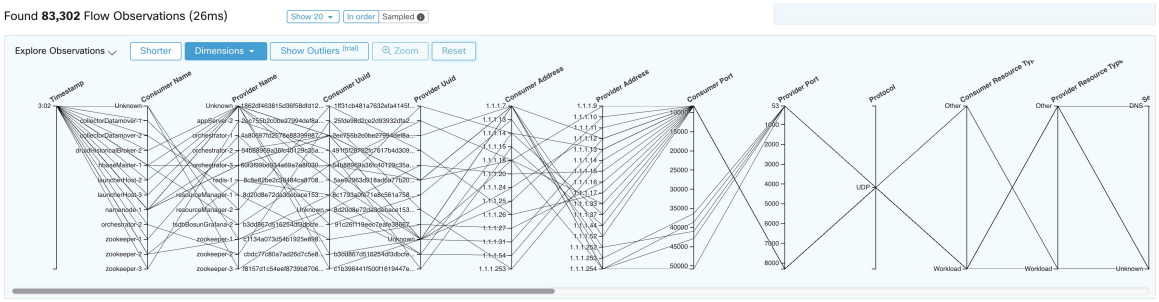
さらに、[順方向/逆方向ファブリックパス（Fwd/Rev Fabric Path）] に関する詳細が利用可能になります。各リンクをクリックして、[レイテンシ（Latency）] および [ドロップインジケータ（Drop Indicators）] の時系列チャートの切り替えができます（ゼロでない場合）。[順方向（Fwd）] または [逆方向（Rev）] をクリックすると、フローの [ファブリックパスのオーバーレイ（Fabric Path Overlay）] ページのドリルダウンに移動します。

図 *392 :* ファブリックパス



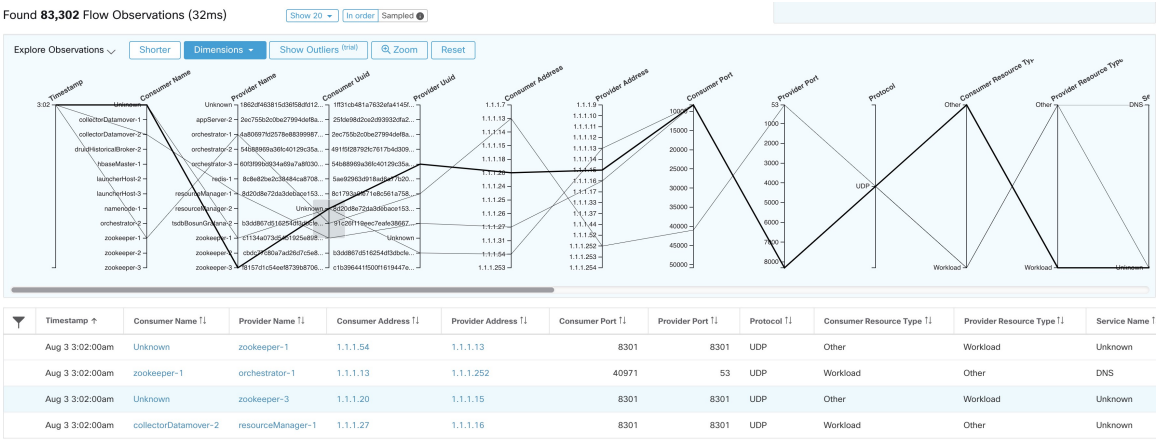
観測結果の確認

図 *393 :* 観測結果の確認



青色の [観測結果の確認（Explore Observations）] ボタンをクリックすると、チャートビューが有効になり、高次元データ（別名「平行座標」チャート）をすばやく分析できます。最初は少し難しく感じるかもしれませんが、関心のある次元のみを有効にする場合（[次元（Dimensions）] ドロップダウンの項目のチェックを外す）や、次元の順序を並べ替える場合、このチャートは非常に便利です。このチャートの 1 本の線は 1 つの観測値を表し、その線がさまざまな軸と交差する場所は、その次元での観測値を示しています。より明確にするには、チャートの下にある観測結果のリストにカーソルを合わせます。これにより、チャート内の観測結果を示す線が強調表示されます。

図 *394* : フローの観測結果にカーソルを合わせた場合



Timestamp	Consumer Name	Provider Name	Consumer Address	Provider Address	Consumer Port	Provider Port	Protocol	Consumer Resource Type	Provider Resource Type	Service Name
Aug 3 3:02:00am	Unknown	zookeeper-1	1.1.1.54	1.1.1.13	8301	8301	UDP	Other	Workload	Unknown
Aug 3 3:02:00am	zookeeper-1	orchestrator-1	1.1.1.13	1.1.1.252	40971	53	UDP	Workload	Other	DNS
Aug 3 3:02:00am	Unknown	zookeeper-3	1.1.1.20	1.1.1.15	8301	8301	UDP	Other	Workload	Unknown
Aug 3 3:02:00am	collectorDatamover-2	resourceManager-1	1.1.1.27	1.1.1.16	8301	8301	UDP	Workload	Other	Unknown

Download table data as JSON

フローデータの高次元の性質により、このチャートはデフォルトでかなり幅が広く、チャート全体を表示するには右にスクロールする必要があります。このため、関心のある次元以外はすべて無効にすると便利です。

サンプリング観測と順次観測の比較

[観測結果の確認（Explore Observations）]では、**サンプリング**を有効にして、より多くのフローを対象にすることを推奨します。これにより、選択した期間内に発生したさまざまなフローをより多く確認できます。したがって、上の時系列チャートで200万件のフロー観測を選択した場合、1000件のサンプルが期間全体で均一にロードされます。一方、フローの**順次**ロードでは、期間の最初から1000件のフロー観測データがロードされます。：

図 *395* : *1000* 件の順次データ

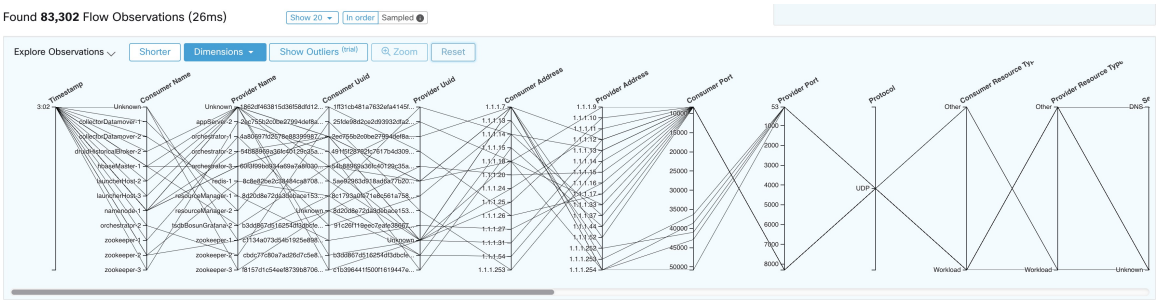
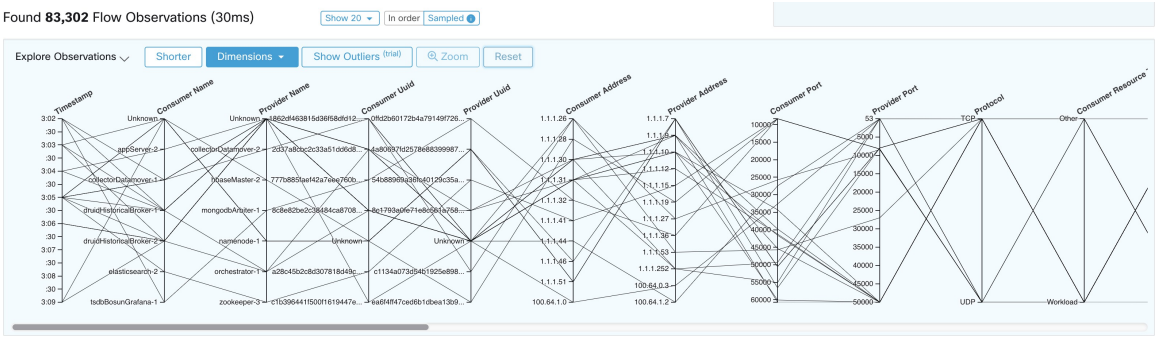


図 *396 : 1000* 件のサンプルデータ

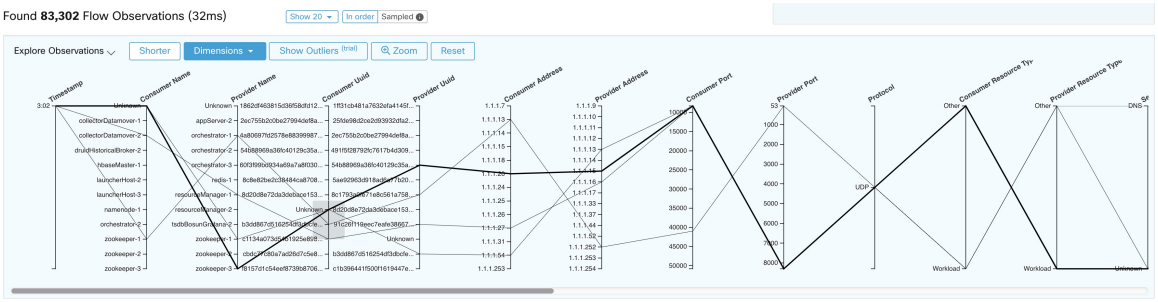


順序どおりの観測データの場合は、すべて**タイムスタンプ**が 9:09 から始まる点や、サンプリングの場合は観測データが選択された期間全体で均等に分散されている点に注目してください。

フィルタリング

いずれかの軸に沿ってカーソルをドラッグすると、選択範囲が作成され、その選択範囲に一致する観測データのみが表示されるようになります。軸を再度クリックすると、いつでも選択範囲を解除できます。一度に任意の数の軸を選択できます。観測データのリストが更新され、選択した観測データのみが表示されます。

図 *397 :* 選択範囲による確認



Timestamp	Consumer Name	Provider Name	Consumer Address	Provider Address	Consumer Port	Provider Port	Protocol	Consumer Resource Type	Provider Resource Type	Service Name
Aug 3 3:02:00am	Unknown	zookeeper-1	1.1.1.54	1.1.1.13	8301	8301	UDP	Other	Workload	Unknown
Aug 3 3:02:00am	zookeeper-1	orchestrator-1	1.1.1.13	1.1.1.252	40971	53	UDP	Workload	Other	DNS
Aug 3 3:02:00am	Unknown	zookeeper-3	1.1.1.20	1.1.1.15	8301	8301	UDP	Other	Workload	Unknown
Aug 3 3:02:00am	collectorDatamover-2	resourceManager-1	1.1.1.27	1.1.1.16	8301	8301	UDP	Workload	Other	Unknown

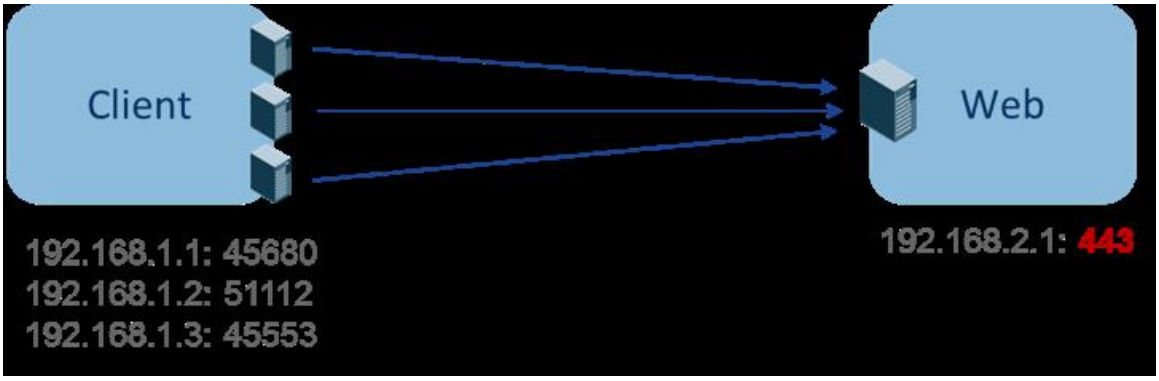
Download table data as JSON

クライアントサーバーの分類

フローの方向（クライアント/サーバーまたはプロバイダー/コンシューマの分類）は、可視性、自動ポリシー検出、および適用にとって重要です。すべてのユニキャストフローには、クライアントとサーバーの分類があります。

たとえば、HTTPS を使用して Web サーバー（192.168.2.1）にアクセスするクライアント（192.168.1.1 ～ 192.168.1.3）がある場合、通常、送信元ポートは 1025 ～ 65535 の範囲のエフェメラルポートであり、宛先ポートは 443 です。

図 *398* : クライアントサーバーの分類



正確なクライアントサーバーの方向は次のとおりです。

- クライアント：192.168.1.1 ～ 3
- サーバー：192.168.2.1
- サービス：TCP ポート 443

自動ポリシー検出によって生成されたポリシーは、以下の図に示されています（左側のエンドポイントがグループ化されています）。

図 *399* : 生成されたポリシー



ここで、クライアントとサーバーの方向の判断が逆になった場合（不正確な分類）、次のようになります。

- クライアント：192.168.2.1
- サーバー：192.168.1.1 ～ 3
- サービス：エフェメラルポートのリスト（45680、51112、45553）

次に、上記の不正確な分類では、生成されたポリシーは次の図のようになる可能性があります。

図 *400 :* 不正確な分類



この結果、ポリシーの適用に関してより多くのリソースが消費されます。さらに、ポリシーの適用方法によっては、192.168.1.1 ～ 3 では当該エフェメラルポートを使用しても、192.168.2.1 にアクセスできません。たとえば、Secure Workload ソフトウェアセンサーの適用を使用する場合、前述のクライアントから Web（ESTAB）への適用ポリシーは、Web 宛てのクライアントによって生成されたトラフィック（NEW、ESTAB）と一致しません。

タイムスタンプと TCP フラグは、クライアントとサーバーの方向を判断するために Secure Workload で使用されます。たとえば、パケットが UDP/ICMP である可能性があるか、方向信号をサポートしていない HW センサーが使用されているため、TCP フラグ情報（SYN、SYN/ACK）がない場合、ユーザー定義のオーバーライドルール、タイムスタンプ、およびその他のヒューリスティックは、フローの方向を推測するために使用されます。定義上、ヒューリスティックでは 100% の精度は保証されません。クライアントとサーバーの精度は、使用されるセンサーのタイプと、センサーの使用条件の関数で得られます。ユーザーは、Cisco Secure Workload の REST-API（OpenAPI）を使用してクライアントとサーバーのオーバーライドルールを挿入し、Secure Workload が誤った方向を取得したフロータイプのサーバーポートを識別できます。次に、Secure Workload はそれらのルールを適用してキャプチャされた新しいフローデータを処理し、フローの方向が固定されている期間にポリシーを生成します。オーバーライドルールを指定する API の詳細については、クライアントサーバー構成（1192 ページ）を参照してください。ポリシーを手動で定義し、不要なポリシーを調査または削除することもできます。ポリシー （1075 ページ） を参照してください。

センサータイプの推奨事項

優れた可視性エージェントまたは適用ソフトウェアエージェントは、Secure Workload クライアントサーバー分類アルゴリズムに最適な信号を提供します。優れた可視性エージェントまたは適用エージェントの展開を検討することを強く推奨します。これらのエージェントは、適切なクライアント/サーバー分類を推進するために必要なすべての信号を取得します。一部のワークロードで優れた可視性エージェントまたは適用エージェントを展開できない場合は、ERSPAN センサーを使用し、そこで停止して自動ポリシー検出を行うことを推奨します。そのためには Secure Workload が役立ちます。また、シスコではフィードバックに基づいてヒューリスティック アルゴリズムを継続的に改善しています。

正しいクライアントサーバーの指示情報が利用できない場合、Secure Workload はユーザー定義のオーバーライドまたはヒューリスティックを使用して、指示の内容を推測します。定義上、ヒューリスティックは 100% の精度を保証しません。使用するセンサーの種類や使用条件により精度は低下します。

以下は、ポリシー生成のユースケースにおけるクライアント/サーバーの決定で推奨される順序です。

- [優れた可視性エージェントまたは適用エージェント （Deep visibility or enforcement agents）]：最良の結果を得るには、ソフトウェアセンサー（優れた可視性エージェントまたは適用エージェント）を使用します。センサーが起動する前に開始されたトラフィックフローは、以下で説明するヒューリスティックによって処理されます。
- [F5/Citrix/. ..エージェントのようなADCセンサー（ADC Sensors like F5/Citrix/. . . agents）]：これらのエージェントは、ADC デバイスからクライアントサーバーの状態を収集し、その信頼できる情報源を Cisco Secure Workload にストリーミングします。
- [ERSPANセンサー（ERSPAN sensors）]：ERSPAN センサーを使用する場合、ユーザーは問題があるワークロードとの間のトラフィックを完全に可視化し、ERSPANセンサーがすべてのスパンされたトラフィックを認識できるようにする必要があります。また、ワークロードのネットワーク通信の可視性を低下させないため、 ERSPAN センサーのオーバーサブスクライブもしないでください。さらに、ERSPANセンサーのパケットドロップを最小限に抑える必要があります。オペレータには、自動ポリシー検出のネットワークフロー情報を含むプロセス情報は見えません。

以下にリストされている Netflow センサーを使用している場合、ユーザーはポリシー分析に関するより多くの手動作業にサインアップし、例外ルールを生成する必要があります。定義上、Secure Workload は 100% 正確ではないヒューリスティックを多用します。

- [Netflowセンサー（Netflow Sensor）]：NetFlow は、サンプリングされ、集約されたフローデータを提供します。集約およびサンプリングのプロセスでは、クライアントサーバーの指示情報が失われます。指示情報が失われると、自動ポリシー検出とポリシー生成の結果に影響し、問題をより困難にします。NetFlow データは、高度な可視性に優れています。Secure Workloadはヒューリスティックにフォールバックする必要がありますが、正しくない場合はオペレータに代わってより多くの手動作業が必要になります（たとえば、Cisco Secure Workload の例外ルールを定義するなど）。NetFlow データも一部の短いフローを見逃しており、信号品質は NetFlow データを生成するデバイスに依存します。アプリケーション デリバリ コントローラ（またはサーバーロードバランサ）など、L3/L4 NAT デバイスを介したスティッチングフローのような特殊なユースケースでは、Secure Workload で NetFlow を使用して、どのフローが他のどのフローに関連しているかを Secure Workload で可視化することを推奨します。

クライアントサーバーの指示分析の詳細は、次のとおりです。

フローのプロデューサー（別名サーバー）とコンシューマ（別名クライアント）の識別

サーバーを検出するために使用される方法（多くの場合、ヒューリスティック）は複数あります。

- センサーが SYN ハンドシェイクを検出すると、サーバーを特定できます。

- 時間ベース：接続のイニシエータがクライアントと見なされます。
- 度合いモデル：通常、サーバーは多くのクライアントと通信します。対照的に、クライアントポートの利用度合いは、はるかに少ないと予想されます。

優先順位は、SYN_ANALYSIS/NETSTAT > USER_CONFIG > DEGREE_MODEL の順です。

SYN_ANALYSIS にユーザー設定よりも高い優先順位が与えられるのは、ユーザー設定は古くなる可能性があり、センサーがグラウンドトゥルースを確立するための最良の監視ポイントを持っていると考えられるからです。DEGREE_MODEL は学習/ヒューリスティックが機能する場所であり、精度を 100% 保証することはできません。

クライアントサーバー検出のヒューリスティックは、この分野でシスコが最善を尽くし、継続的にアルゴリズムを改良しているにも関わらず、うまく機能しない場合があります。こうした場合は、OpenAPIインターフェイスを使用して、既知のサーバーポートにアクセスできます。この設定は過去のフローには適用されず、その時点以降（つまり、今後）のフローのマーキングにのみ影響します。これは、通常の対処方法ではなく、フォールバックの最後の手段として使用できます。

また、特定のフローの全期間中、クライアントサーバーのマーキングを繰り返し切り替えないようにすることも重要です（間違っていたり、内部モデルが変更された場合でも、フローパターンがより多く観測/分析されるにつれて、時間の経過とともに変化します）。同等以上の優先順位の更新により、低い優先順位の更新を上書きできます（既存のフローのクライアントサーバーも切り替わります）。言い換えれば、「フローの存続期間中」のマーキングの粘着性は、度合いモデルベースのマーキングにのみ適用されます。

カンバセーションモード

デフォルトでは、エージェントのフロー分析忠実度モードは「詳細」です。従来は、これが利用可能な唯一のモードであり、観測されたすべてのフローがそれらのフローに関する詳細な統計とともにエージェントによってレポートされていました。統計には、パケット数とバイト数、TCP フラグ、接続統計、ネットワーク遅延、SRTT などが含まれます。

この種のレポートは多くの場合望ましいものですが、レポートと処理のための計算負荷が大きくなります。また、主要なユースケースがセグメンテーションのみの場合は厳密な要件ではない可能性があります。

カンバセーションモードは、従来の詳細モードよりも軽量な代替モードになります。カンバセーションモードのエージェントは、可能な限り（つまり、クライアントとサーバーを正確に分類できる場合はいつでも）フローではなくカンバセーションをレポートすることを目指します。これは、TCP、UDP、および ICMP フローに適用されます。

詳細モードでは、TCP/UDP フローの場合、5 タプルフロー（送信元と宛先の IP、送信元と宛先のポート、およびプロトコル）をレポートします。

一方で、カンバセーションモードでは、送信元ポートは（新しく接続されるたびに変更される）エフェメラルポートであるため、エージェントは送信元ポートを省略し、4 タプルフローにします。

（注） フローを4タプルとして検出することは、クライアントサーバー検出アルゴリズムにも依存しています。このアルゴリズムは、サーバー/宛先ポートがウェルノウンポート（0 ～ 1023）であることに依存しています。

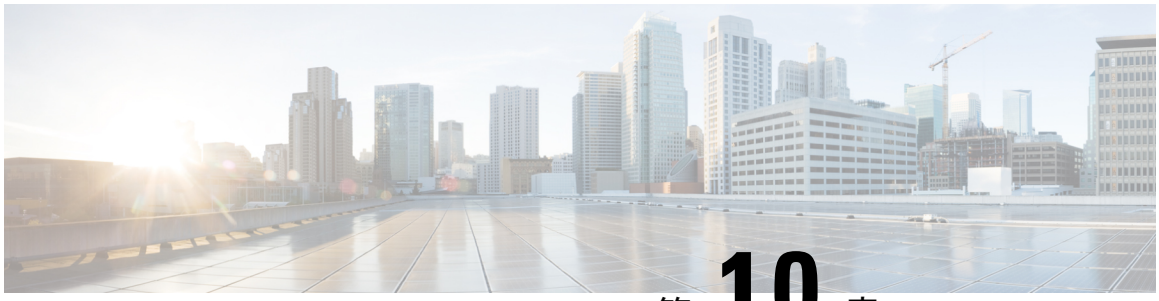
そのため、ウェルノウンサーバー/宛先ポートを使用しないカスタムアプリケーションを使用している場合は、OpenAPIインターフェイスを使用してウェルノウンサーバーポートにアクセスできます。この設定は過去のフローには適用されず、その時点以降（つまり、今後）のフローのマーキングにのみ影響します。サーバーポートを最適化するには、「Client Server Configuration」を参照してください。

カンバセーションモードのエージェントレポートには、トリミングされた情報が含まれています。省略されたフィールドの完全なリストは次のとおりです。TCP/UDP送信元ポート（エフェメラルポート）、順方向/逆方向TCPボトルネック、TCPハンドシェイクバケット、SRTT（マイクロ秒）、順方向/逆方向パケット再送信、利用可能なSRTT、削減された順方向/逆方向輻輳ウィンドウ、変更された順方向/逆方向MSS、順方向/逆方向TCP受信Window Zero、順方向/逆方向バーストインジケータ、順方向/逆方向最大バーストサイズ（KB）。

カンバセーションモードを有効にするには、「ソフトウェアエージェントの設定」に含まれる「フローの可視性設定」セクションを参照してください。

（注） エージェントをカンバセーションモードでレポートするように変更することで得られる確実な利点は、TCPフローのパーセンテージ、既定のサービスポートでリッスンするサービスの数、エージェントのメモリ制限など、複数の要因によって異なる場合があります。

（注） 一部のエージェントの「カンバセーション」モードをオンにすると、フロー検索ページの観測結果にカンバセーションとフローが混在する場合があります。



第 10 章

アラート

図 *401* : アラートタイプとパブリッシャの設定

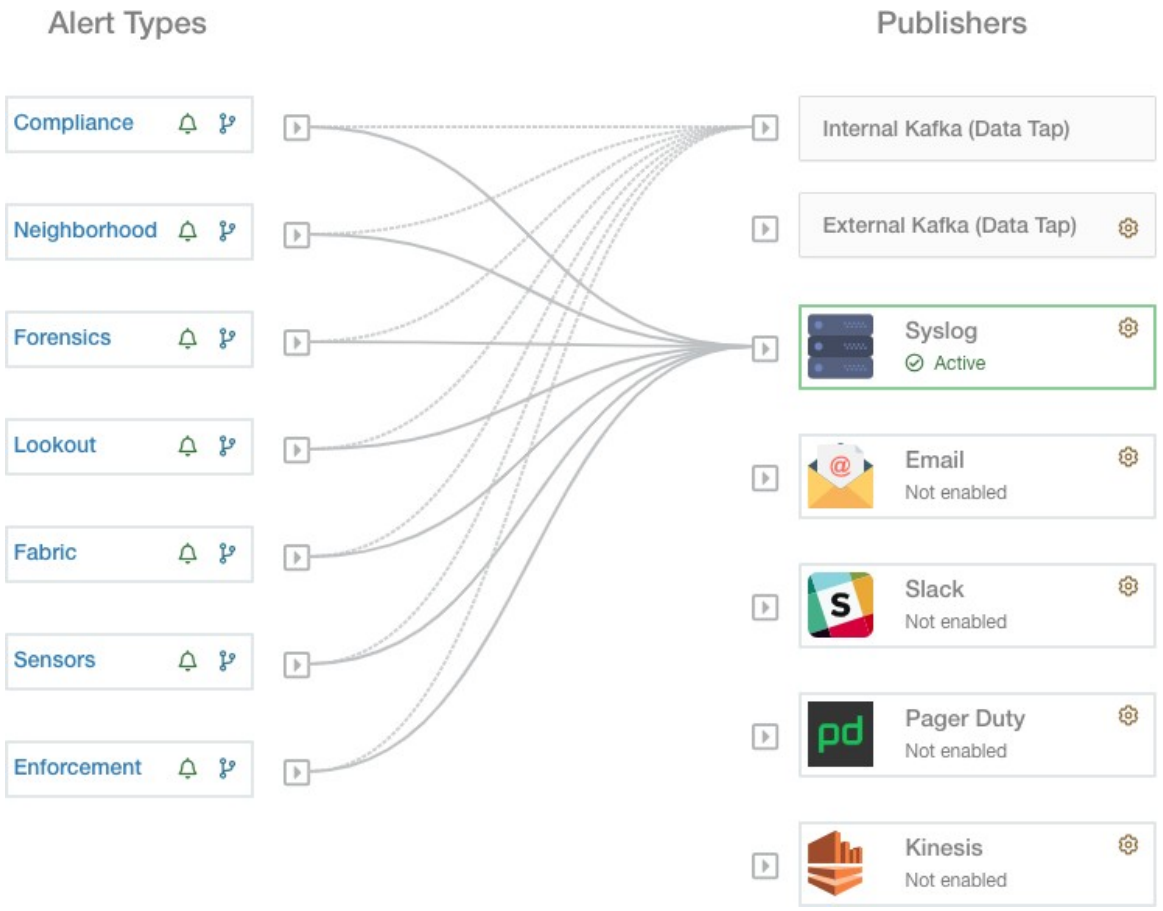
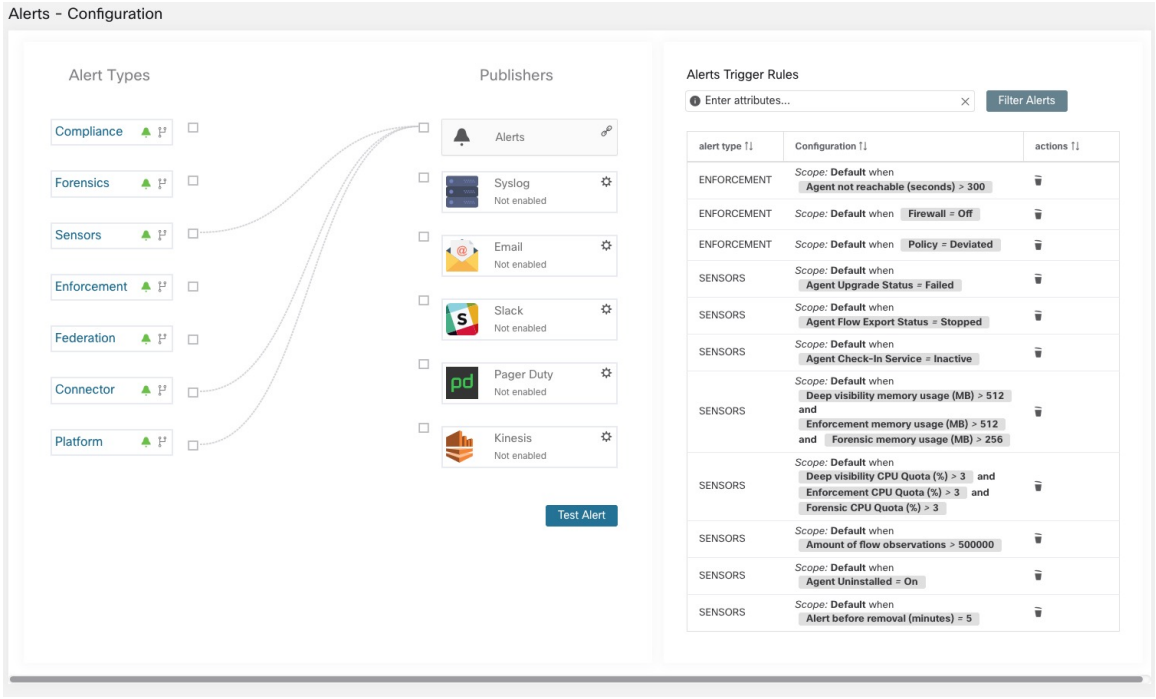


図 *402 :* アラートタイプとパブリッシャの設定



Cisco Secure Workload 内のアラートは、多くの統合コンポーネントで構成されています。コンポーネントは、大まかに次のように分類できます。

Visibility:

- [アラート（Alerts）] ページ：**[Investigate]** > **[アラート（Alerts）]**にあります。このページは、DataTap に送信されたアラートのプレビューで構成されます。

アラートの送信元と設定：

- [アラート設定（Alert Configuration）]：アプリケーション/コンポーネントによって判断されますが、多くの場合、DataTap の構成やサマリーアラートオプションなどの機能を備えた共通のインターフェイス（「アラート設定モーダル」と呼ばれる）が使用されます。
- [アラート設定（Alerts Configuration）] ページ：**[管理（Manage）]** > **[アラート設定（Alerts Config）]**にあります。このページには、共通のモーダルを使用して構成されたアラート設定と、アラートパブリッシャおよび通知者の設定の両方が表示されます。

アラートの送信先：

- アラートアプリケーション：生成されたアラートを設定済みの DataTap に送信する暗黙的な Secure Workload アプリケーション。アラートアプリケーションは、スヌーズやミュートなどの機能を処理し、実質的に**送信**するアラートを判断します。
- アラートパブリッシャ：UI に表示されるアラートの数を制限し、外部で使用できるようにアラートを Kafka（MDT または DataTap）にプッシュします。

- Edge アプライアンス：Slack、PagerDuty、Email などの他のシステムにアラートをプッシュします。

- アラートの設定 （781 ページ）
- 現在のアラート （798 ページ）
- アラート詳細 （803 ページ）

アラートの設定

[アラート設定（Alerts Configuration）]ページでは、アラートトリガールールを設定し、アラートを送信するパブリッシャを選択できます。このページに表示されるアラートタイプは、ユーザーロールによって異なります。アラートのパブリッシャは、Kafka（データタップ）または Notifier のいずれかです。

[アラート設定（Alerts Configuration）]ページでは、アラートトリガールールを設定し、アラートを送信するパブリッシャを選択できます。このページに表示されるアラートタイプは、ユーザーロールによって異なります。アラートのパブリッシャは、アラートまたは Notifier のいずれかです。

（注） アラートとコンプライアンス アプリケーションは、リリース 3.0 以降、Secure Workload App Store から削除されます。アラートアプリインスタンスやコンプライアンス アプリ インスタンスを作成しなくても、このページでコンプライアンスアラートをはじめとするアラートを設定できます。

アラートの作成

図 *403 :* アラート（トリガールール）の作成

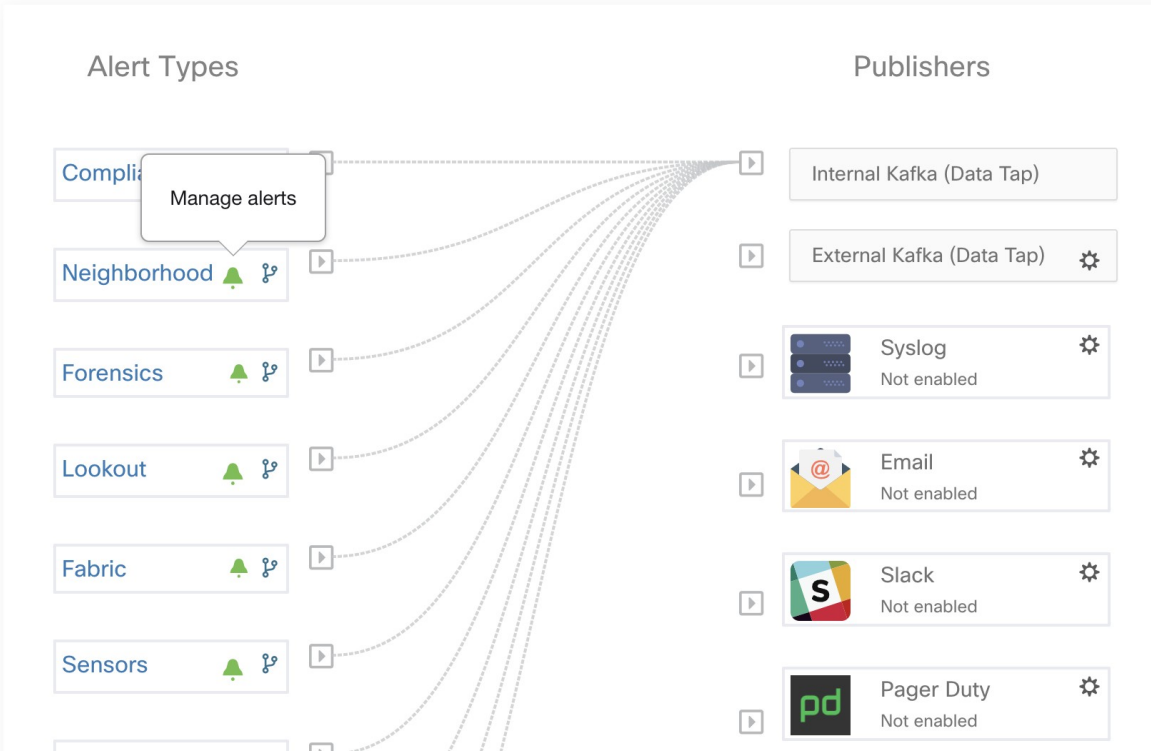
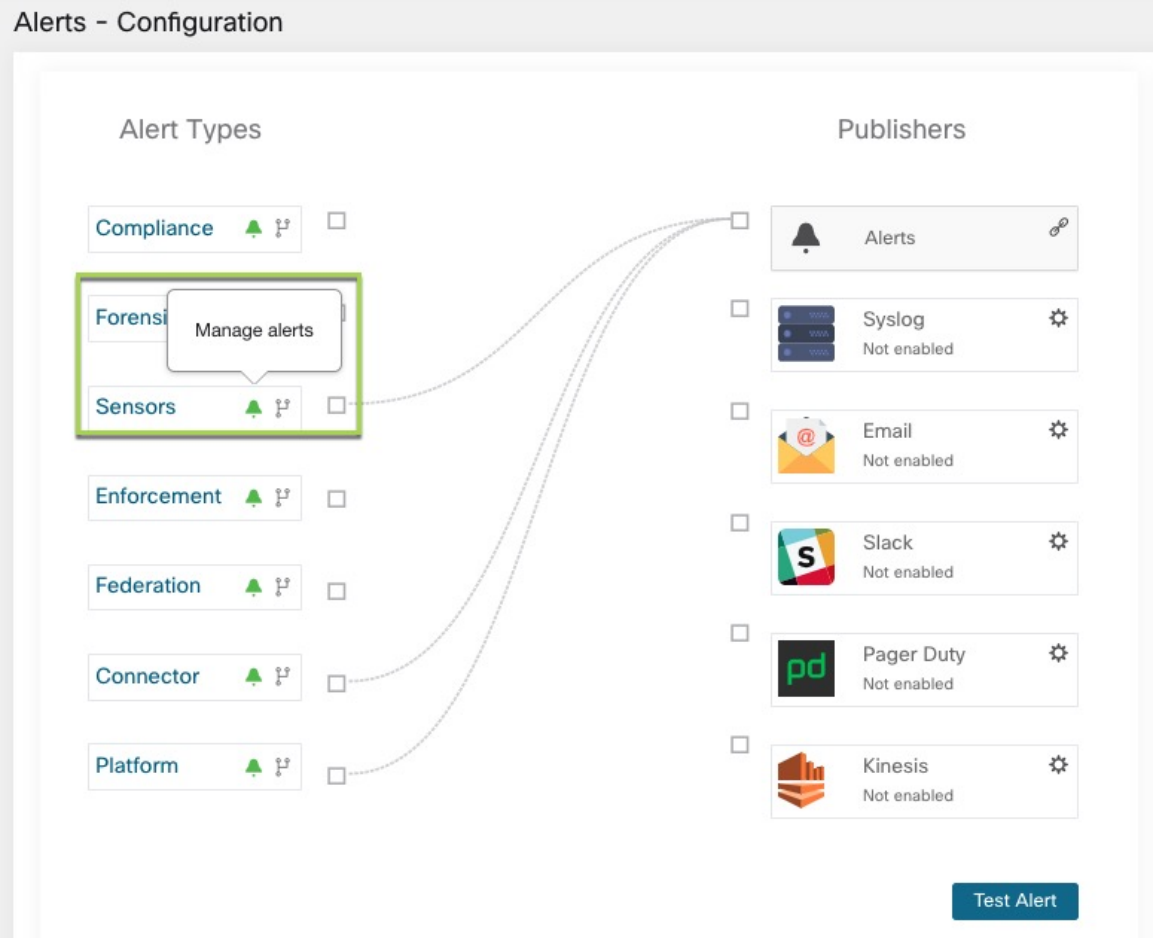


図 *404* : アラート（トリガールール）の作成



いくつかのコンポーネントは、アラートを設定するため、共通のアラート設定モーダルを使用します。現時点では、設定モーダルには次のものが含まれます（特定のアラート設定の詳細については、各ユーザーガイドを参照してください）。

- 近隣
- 適用アラート
- センサーアラート
- 適用アラート
- センサーアラート

（注） コンプライアンス アラート タイプの場合、現在選択されている範囲で最低限「適用」機能を使用可能なユーザーのみが、アラートトリガールールを作成できます。

（注） 適用およびセンサーアラートタイプの場合、アラートトリガールールは、現在選択されているルート範囲に適用されます。

次のタイプには、設定モーダルがありません。

- フォレンジックアラート： フォレンジックルールを使用して設定
- コネクタアラート

アラート設定モーダル

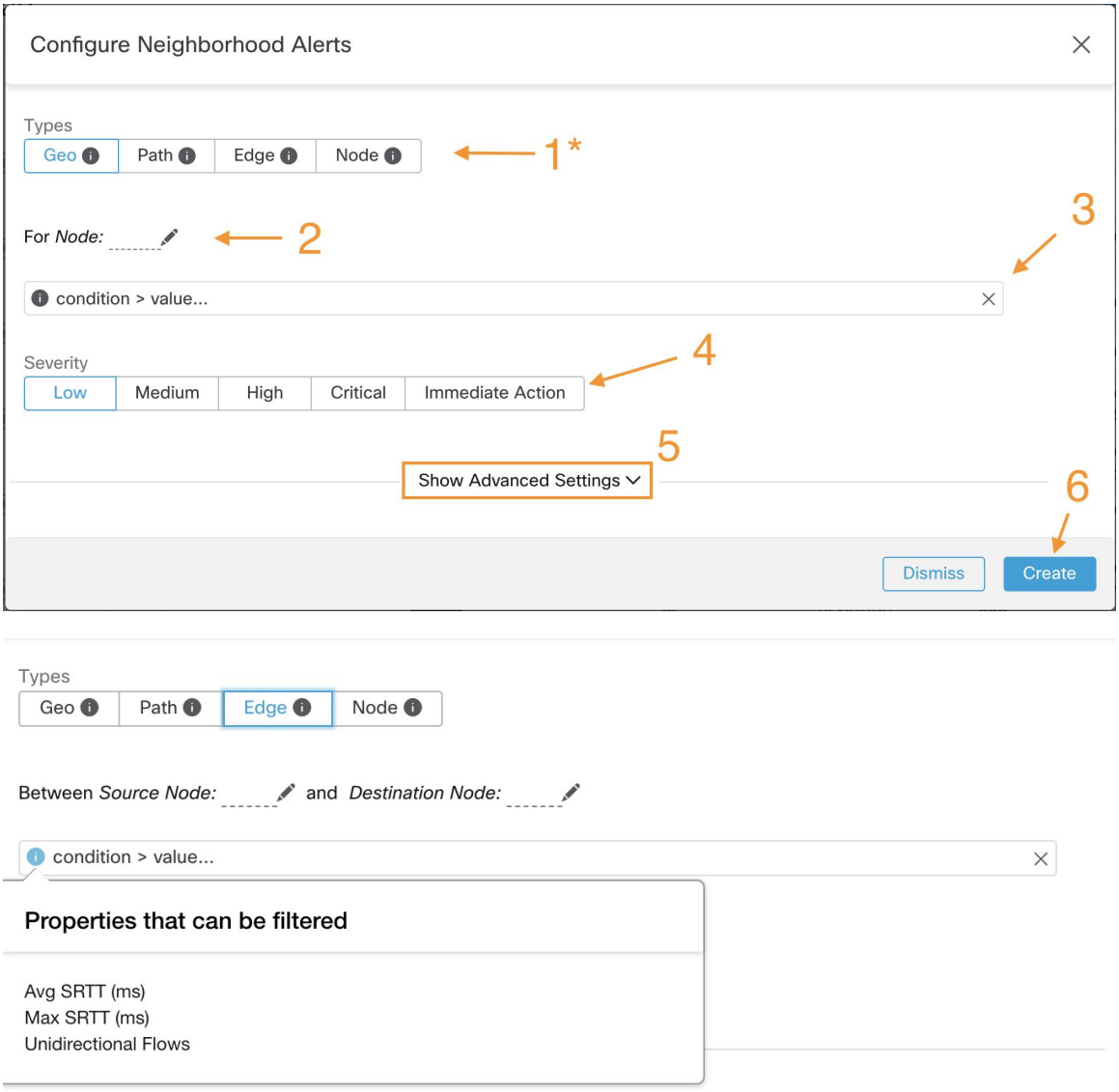
アラート設定モーダルは、次の 6 つのセクションで構成されています。

1. アラートのタイプ。注：これは、アラートの設定が件名に応じて異なる場合にのみ表示されます（現在、近隣アラートに対してのみ表示されます）。
2. アラートの件名:「何をアラートするか」 を示します。これはアプリによって異なり、アラートモーダルがコンテキスト型の場合は事前に入力されている可能性があります。
3. アラートがトリガーされる条件:「いつアラートを生成するか」を示します。にカーソルを合わせると、指定できる条件の一覧が表示されます。注：この一覧には、現在設定されているアラートタイプに特化した条件が表示されます。
4. アラートのシビラリティ（重大度）の選択。多くのアラートが生成された場合、シビラリティ（重大度）の高いアラートが、重大度の低いアラートよりも優先的に UI に表示されます。
5. サマリーアラートオプションで構成される追加の設定オプション。[詳細設定を表示する（Show Advanced Settings）] をクリックして展開します。
6. モーダルを閉じる：新しいアラートと指定したすべての設定オプションを追加する場合は、[作成（Create）] を選択します。または、新しいアラートを追加しない場合は [取り消し（Dismiss）] を選択します。

1. アラートのタイプ。注：これは、アラートの設定が件名に応じて異なる場合にのみ表示されます。
2. アラートの件名:「何をアラートするか」 を示します。これはアプリによって異なり、アラートモーダルがコンテキスト型の場合は事前に入力されている可能性があります。
3. アラートがトリガーされる条件:「いつアラートを生成するか」を示します。にカーソルを合わせると、指定できる条件の一覧が表示されます。注：この一覧には、現在設定されているアラートタイプに特化した条件が表示されます。

4. アラートのシビラリティ（重大度）の選択。多くのアラートが生成された場合、シビラリティ（重大度）の高いアラートが、重大度の低いアラートよりも優先的に UI に表示されます。
5. サマリーアラートオプションで構成される追加の設定オプション。[詳細設定を表示する（Show Advanced Settings）] をクリックして展開します。
6. モーダルを閉じる：新しいアラートと指定したすべての設定オプションを追加する場合は、[作成（Create）] を選択します。または、新しいアラートを追加しない場合は [取り消し（Dismiss）] を選択します。

図 *405* : アラート設定モーダル

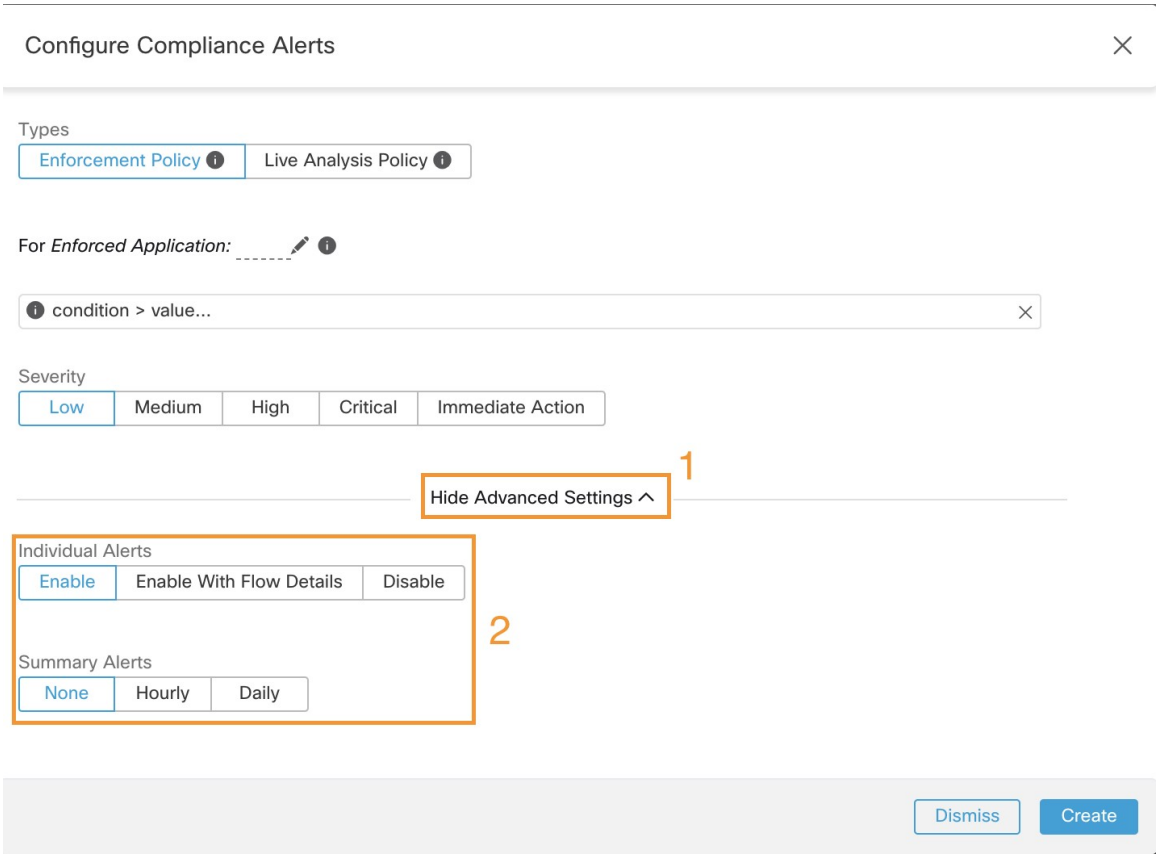


にカーソルを合わせると、アラートトリガーを作成するために使用できるプロパティのリストが表示されます。このリストは、選択したアラートのタイプに依存するコンテキストです。

[詳細設定を表示する（Show Advanced Settings）] をクリックすると表示される追加の設定オプション：

1. [詳細設定を表示しない（Hide Advanced Settings）] をクリックすると、展開が折りたたまれます。
2. サマリーアラートオプション（利用可能な場合）。アラートを生成するアプリによって、利用できるかどうかが決まります。詳細については、「サマリーアラート」を参照してください。

図 *406*：アラート設定モーダルの詳細オプション



サマリーアラート

一部のアプリケーションではサマリーアラートが許可されており、設定オプションはアプリケーションによって異なります。

- 個別のアラートとは、一般的に、集約されていない（または最小限に集約された）情報に対して生成されるアラートを指し、時間範囲は1分間である可能性が高いです。ただし、

アラートが必ずしも1分間隔で実際に生成され、送信されることを意味するわけではないことに注意してください。個別のアラートは引き続き [アプリの頻度（App Frequency）] 間隔で生成されます。

- サマリーアラートとは、1時間の間に生成されたメトリックに対して生成されたアラート、または頻度の低いアラートのサマリーを指します。

App	アプリの頻度 1	個別のアラート	毎時アラート	毎日のアラート
コンプライアンス	毎分	はい：アプリの頻度で	個別のアラートのサマリー	個別のアラートのサマリー
近隣	Hourly	—	対応	毎時アラートのサマリー
施行	毎分	はい：アプリの頻度で	個別のアラートのサマリー	個別のアラートのサマリー
Sensor	毎分	はい：アプリの頻度で	個別のアラートのサマリー	個別のアラートのサマリー

App	アプリの頻度 1	個別のアラート	毎時アラート	毎日のアラート
コンプライアンス	毎分	はい：アプリの頻度で	個別のアラートのサマリー	個別のアラートのサマリー
施行	毎分	はい：アプリの頻度で	個別のアラートのサマリー	個別のアラートのサマリー
Sensor	毎分	はい：アプリの頻度で	個別のアラートのサマリー	個別のアラートのサマリー



（注） サマリーアラートの UI に表示されるイベント時間は、過去 1 時間または指定された間隔ウィンドウにおける同じタイプのアラートの最初の発生時間を表します。

自動要約とスヌーズの違いに関する注記

自動要約はアラート設定に従って生成された全アラートに適用され、スヌーズは特定のアラートに適用されます。アラート設定が非常に具体的である場合、この差ははわずかですが、アラート設定が広範囲である場合、差は大きくなります。

- たとえば、コンプライアンス設定は非常に幅広く、アプリケーションのワークスペースからアラート生成の対象にする違反タイプに関するものまであります。したがって、自動要約は「エスケープ」条件によってトリガーされたすべてのアラートに適用されますが、スヌーズは非常に特定のコンシューマ範囲、プロバイダー範囲、プロバイダーポート、プロトコル、およびエスケープ条件に適用されます。

- 反対に、送信元範囲と宛先範囲間のパスでホップカウントが一定数未満の場合に警告するように設定された近隣アラートは、非常に特定のアラートを生成します。
- たとえば、コンプライアンス設定は非常に幅広く、アプリケーションのワークスペースからアラート生成の対象にする違反タイプに関するものまであります。したがって、自動要約は「エスケープ」条件によってトリガーされたすべてのアラートに適用されますが、スヌーズは非常に特定のコンシューマ範囲、プロバイダー範囲、プロバイダーポート、プロトコル、およびエスケープ条件に適用されます。
- 反対に、送信元範囲と宛先範囲間のパスでホップカウントが一定数未満の場合に警告するように設定されたプラットフォームアラートは、非常に特定のアラートを生成します。

その他の違い

- スヌーズ間隔が経過した後に新しいアラートが生成される際、スヌーズで生成されるのは送信するアラートのみです。スヌーズ間隔中に抑制された可能性のあるアラートの数は示されません。
- 特定の間隔でアラートが生成されている限り、アラートサマリは指定した頻度で生成されます。アラートサマリでは、期間内でトリガーされたアラート数が、集計メトリックや範囲メトリックとともに示されます。

Cisco Secure Workload Alerts Notifier (TAN)

（注） リリース 3.3.1.x では、TAN に代わって **Cisco Secure Workload Edge アプライアンス**が使用されています。

Alerts Notifier は、現在選択されている範囲で Amazon Kinesis、電子メール、Syslog、Slack などの各種ツールを介してアラートを送信する機能を備えています。必要な資格情報や通知アプリケーションに固有のその他の情報を使用して、それぞれの通知者を範囲の所有者やサイト管理者として設定できます。

通知機能の設定

通知機能を設定するには、アラート関連のコネクタを設定する必要があります。このコネクタは、Secure Workload Edge アプライアンスが展開された後にのみ設定できます。Secure Workload Edge アプライアンスの展開方法の詳細については、「コネクタ用の仮想アプライアンス」を参照してください。

Secure Workload Edge アプライアンスをセットアップしたら、各通知機能に固有の必須入力情報を指定して設定できます。Secure Workload Edge アプライアンスのセットアップ後は、アラートタイプと内部 Kafka（データタップ）を結ぶ破線が表示されることに注意してください。これは、通知機能が内部 Kafka（データタップ）に基づいているためです。

Secure Workload Edge アプライアンスをセットアップしたら、各通知機能に固有の必須入力情報を指定して設定できます。Secure Workload Edge アプライアンスのセットアップ後は、アラー

トタイプとアラートのパブリッシャを結ぶ破線が表示されることに注意してください。これは、通知機能がアラートのパブリッシャに基づいているためです。

各アラート通知機能の設定方法の詳細については、「アラート通知用のコネクタ」を参照してください。

[アプリの頻度（App Frequency）]は、アプリが実行されてアラートを生成するおおよその頻度です。たとえば、コンプライアンスには柔軟な実行頻度が設定されており、実際には数分間にわたってアラートを計算する場合があります。

ファブリックアラートは、アプリの実行時に1時間ごとに生成されます（[アプリの頻度（App Frequency）] は 1 時間ごとであることに注意してください）。そのため、個々のアラートオプションで「1 分間」のデータを指定していても、実際のファブリックアラートは「1 時間」のデータの処理が終わったあとでバッチ単位で生成および送信されます。つまり、データが1分あたり 2 個のアラートを生成する場合、実際には 120 個のアラートすべてが 1 時間の最後に生成および送信され、UI にアラートサマリーとして表示される可能性があります。

アラートのパブリッシャの選択

範囲の所有者とサイト管理者は、アラートを送信するパブリッシャを選択できます。パブリッシャには、Kafka（データタップ）と通知ツールがあります。

図 *407* : アラートのパブリッシャの選択

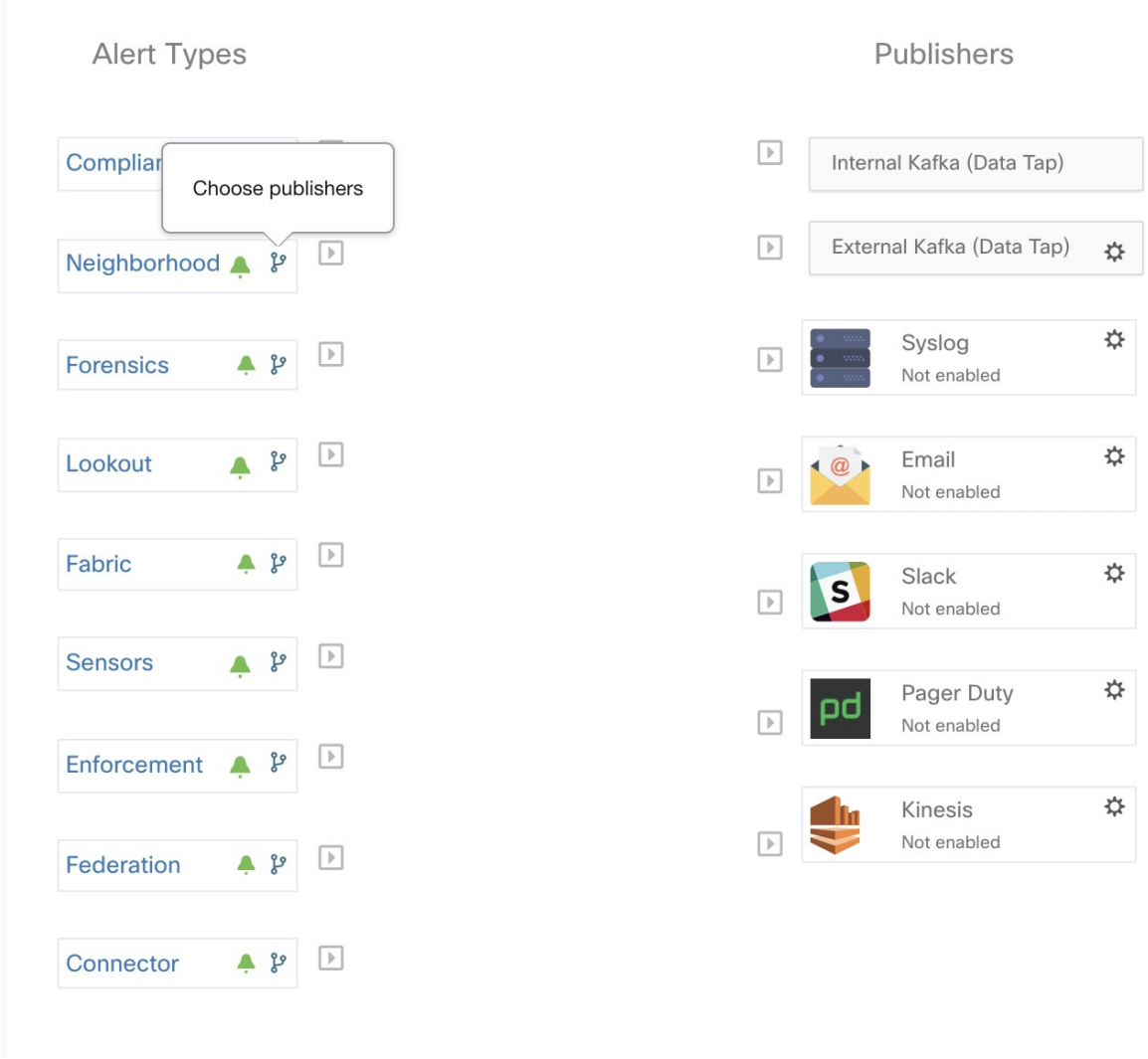
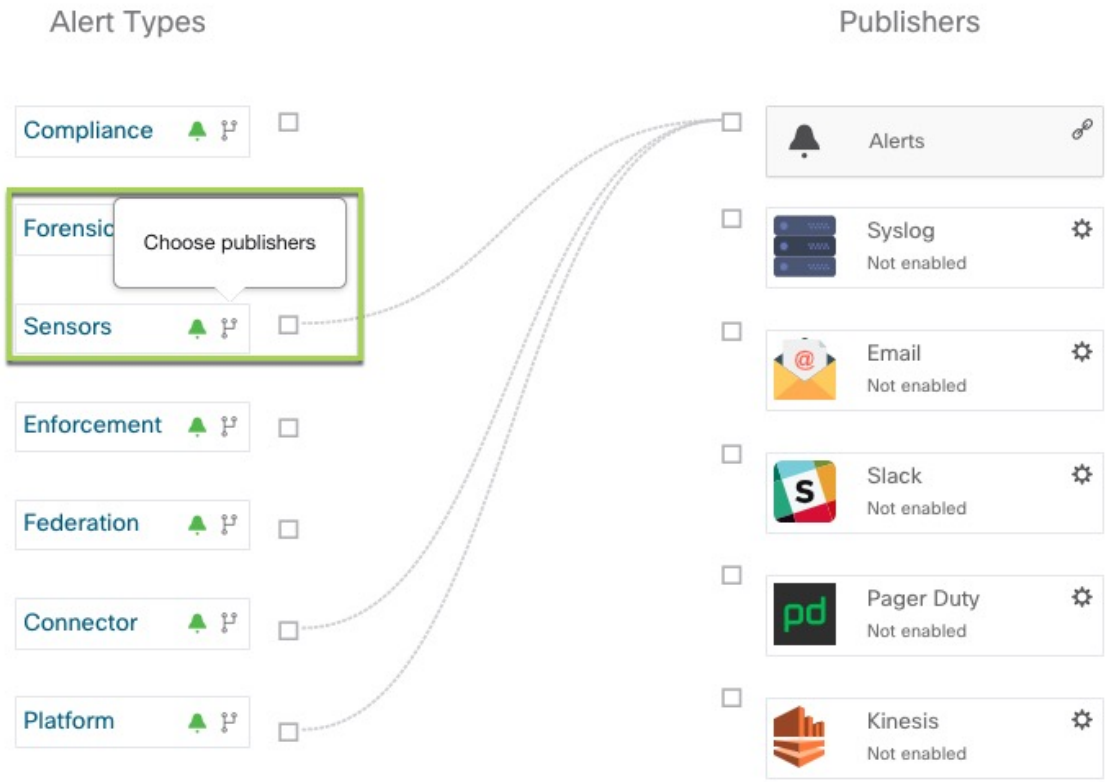
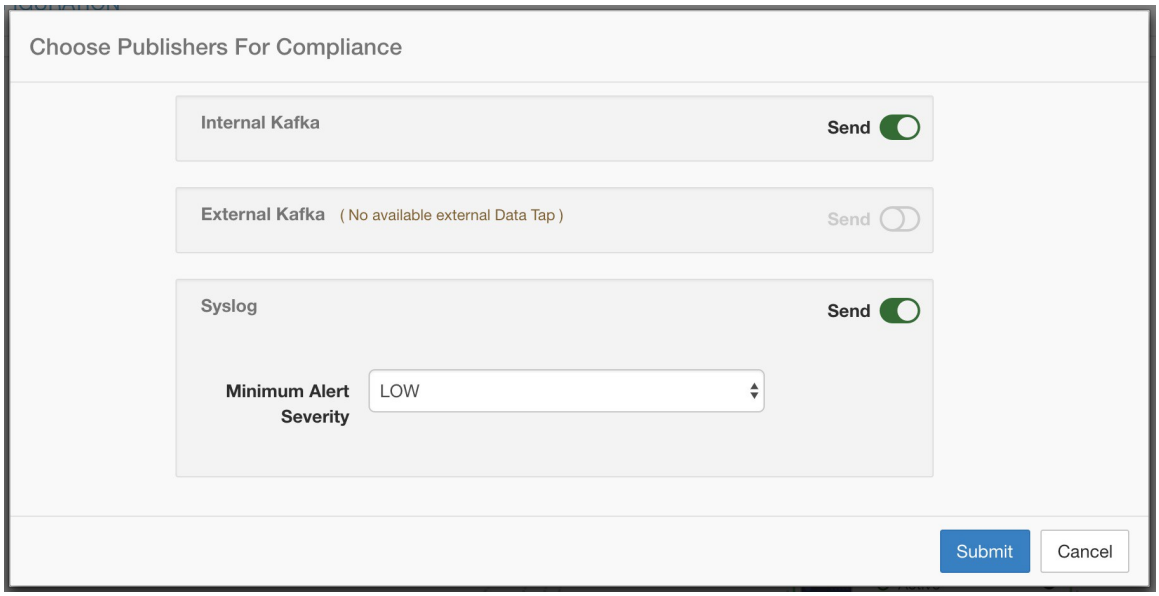


図 *408* : アラートのパブリッシャの選択



（注） サイト管理者と範囲所有者のみが、アラートを送信するパブリッシャを選択できます。

図 *409* : 選択可能なすべてのパブリッシャがこのモーダルに表示されます



選択できるすべてのパブリッシャ（アラートやアクティブな通知ツールなど）がこのモーダルに表示されます。送信ボタンを切り替えて、アラートタイプのパブリッシャを選択できます。アラートの最低シビラティ（重大度）は、重大度レベルを指し、ある特定のアラートがこのレベルに達すると、パブリッシャ経由でアラートが送信されます。

（注） 外部データタップを選択すると、処理できるアラートの最大数に影響を与える可能性があります。処理できるアラートの最大数は、1 分間のバッチあたり最大 14000 アラートまで削減できます。

外部 syslog トンネリングの TAN への移行

（注） 3.1.1.x リリース以降、syslog トンネリング機能は TAN に移行します。プラットフォームレベルの Syslog イベントを取得するために Syslog を設定するには、ユーザーはデフォルトのルート範囲の Secure Workload Edge アプライアンスで TAN を設定する必要があります。デフォルトのルート範囲で Secure Workload Edge アプライアンスを設定したら、syslog サーバーを以下のようにセットアップできます。プラットフォームアラートを有効にするには、プラットフォームの syslog 通知を有効にします。これは、プラットフォーム Syslog 接続を有効にすることで実行できます。

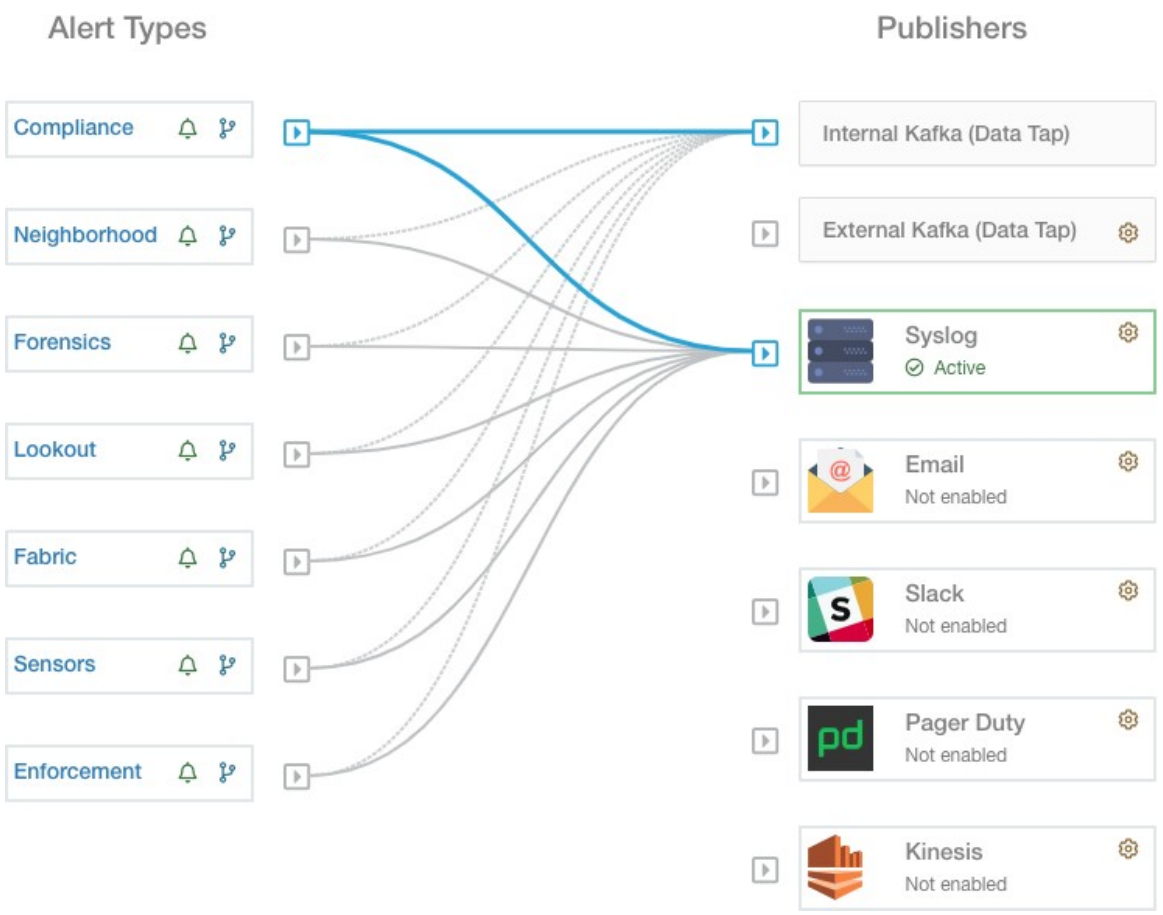
Syslog の設定方法の詳細については、「Syslog Connector」を参照してください。

接続チャート

接続図には、アラートタイプとパブリッシャの間の接続が表示されます。任意のアラートタイプのパブリッシャを選択すると、そのアラートタイプとパブリッシャの間に線が確立されます。内部 Kafka（データタップ）を指す線は、アラート通知構築の基盤となる内部メカニズムを表すため、常に破線であることに注意してください。

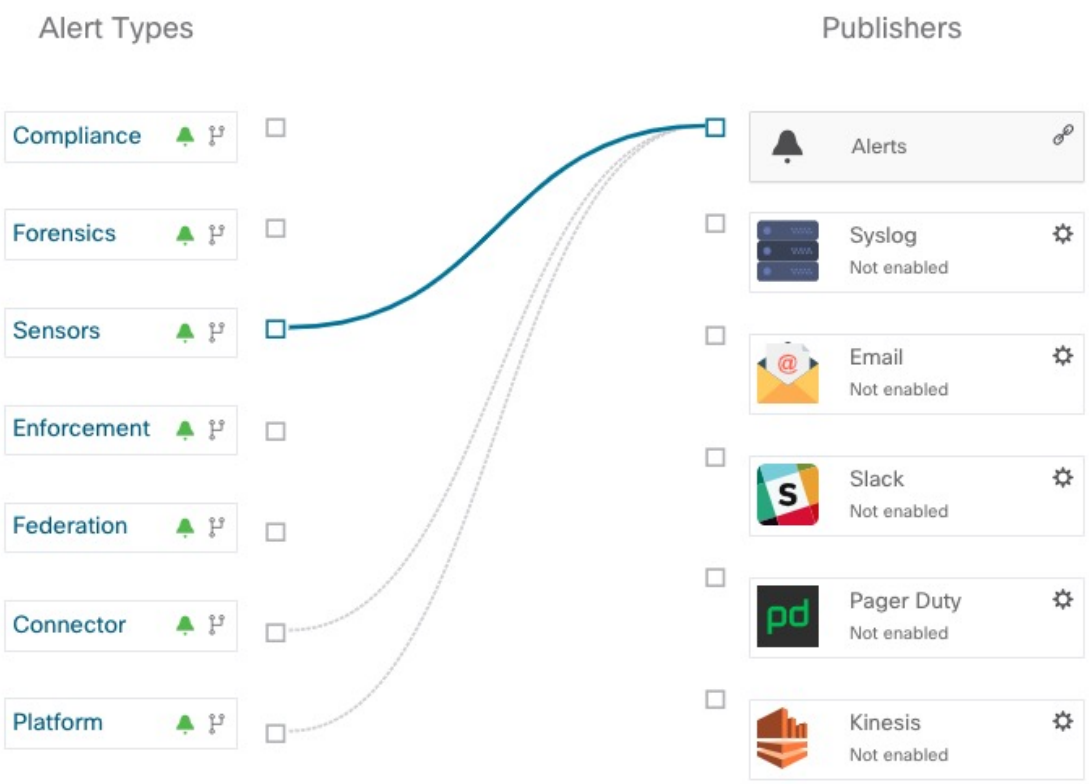
接続図には、アラートタイプとパブリッシャの間の接続が表示されます。任意のアラートタイプのパブリッシャを選択すると、そのアラートタイプとパブリッシャの間に線が確立されます。アラートのパブリッシャを指す線は、アラート通知構築の基盤となる内部メカニズムを表すため、常に破線であることに注意してください。

図 *410* : 接続図



この図に示すように、Syslog が近接アラートのパブリッシャとして選択されると、それらの間に線が確立されます。図の丸で囲まれた領域にカーソルを合わせると、近隣アラートにのみ関連付けられている接続が強調表示されることに注意してください。

図 *411* : 接続図



（注） ユーザーアプリケーションで生成されたアラートは、[アラート設定（Alert Configuration）] ページに表示されません。ユーザーアプリケーションは、構成された任意のデータタップにメッセージとアラートを送信できます。

アラートトリガールールの表示

設定されているすべてのアラートトリガールールのリストが、接続チャートの下の表に表示されます。

図 *412* : アラートトリガールールの表示

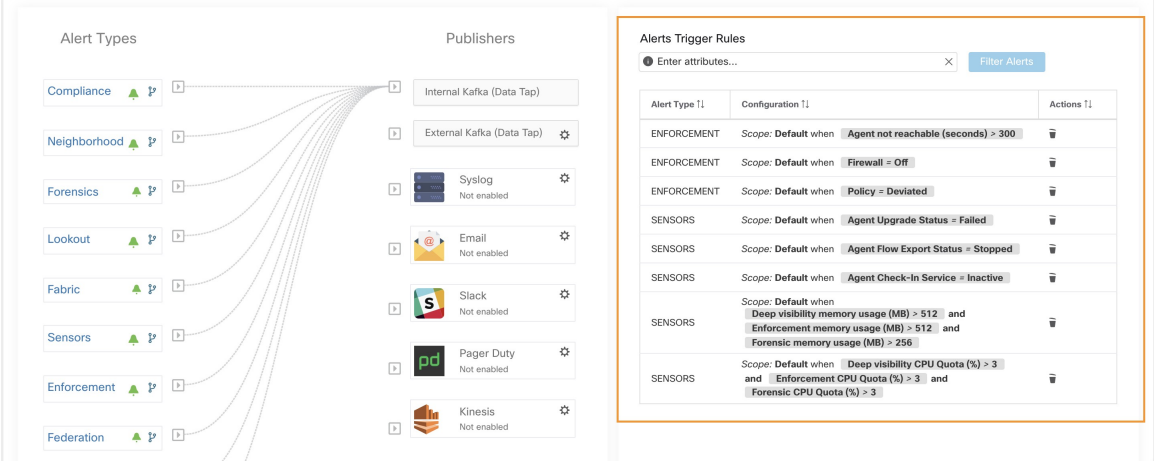
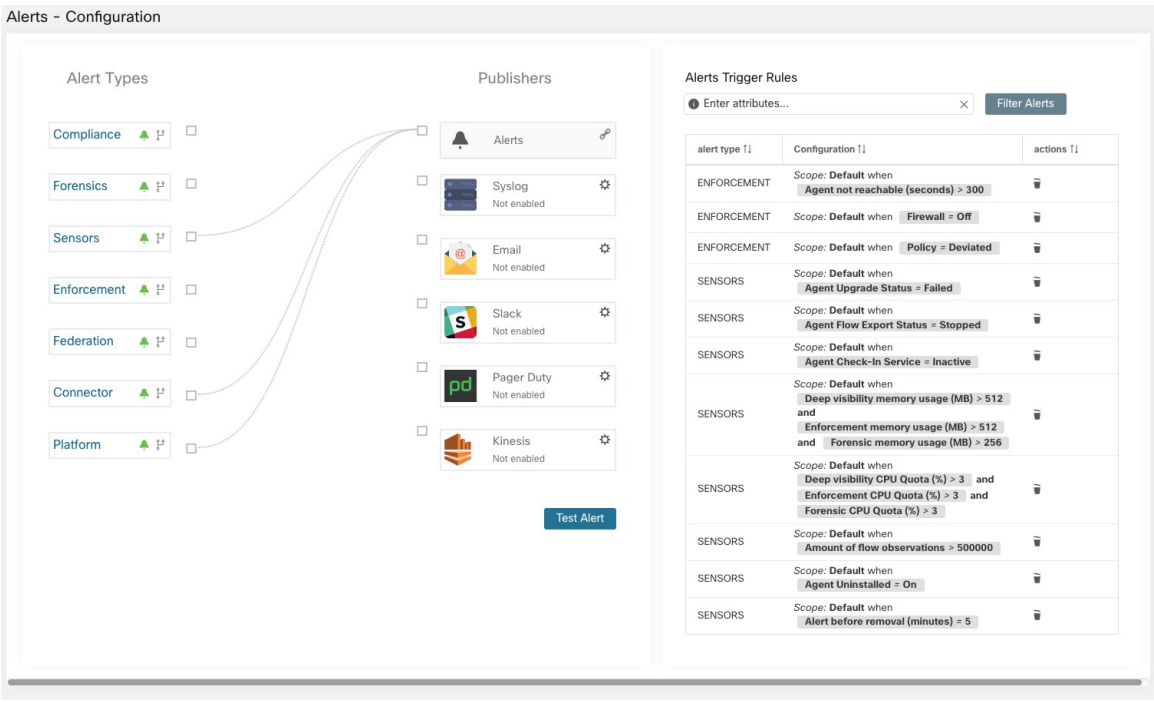


図 *413* : アラートトリガールールの表示



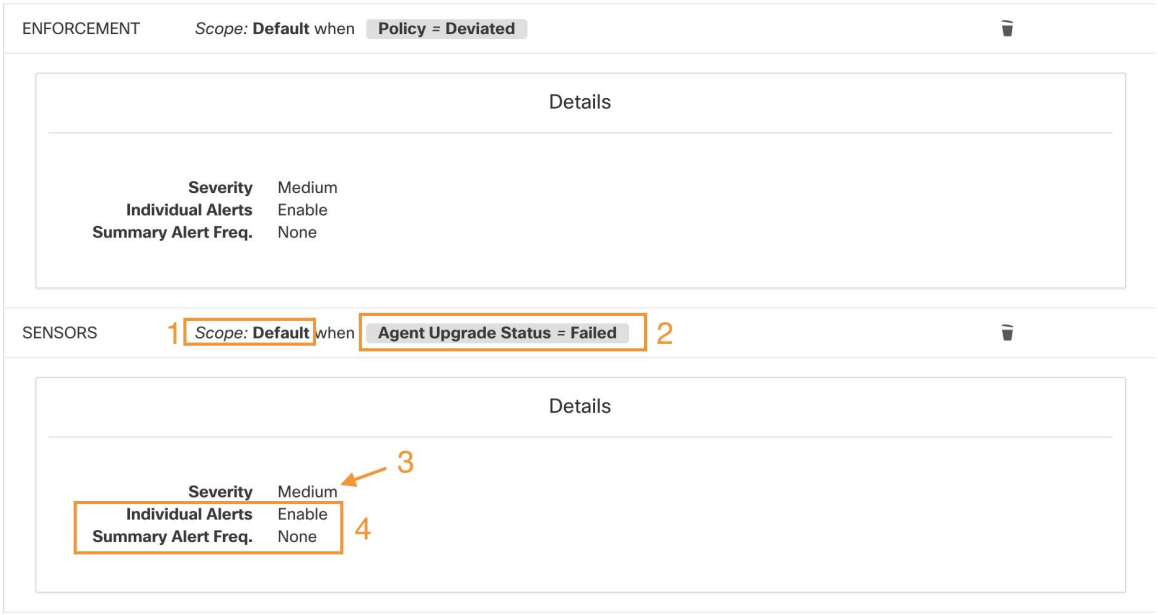
[アラートトリガールール（Alerts Trigger Rules）] テーブルを使用して、アラートタイプ、アラート頻度、およびアラートトリガー条件でアラートトリガールールをフィルタ処理できます。

（注） アラートトリガー条件は完全一致条件です。

アラートトリガールールの詳細

アラート トリガー ルール テーブルの各行をクリックして、設定の詳細を展開できます。

図 414 : 展開されたアラート設定



1. 件名：アラートの内容
2. トリガー：アラートが生成されるタイミング
3. アラートに割り当てられた重大度（同時に生成されたアラートが多くある場合、UIに表示されるアラートに影響する可能性があります）
4. アラートの頻度：個別アラートやサマリーアラートが生成されるかどうか。

テストアラートの生成

サンプルアラートを設定して、アラートの設定のアラートタイプとリンクされたパブリッシャに基づいてアラートを送信することができます。主な用途は、パブリッシャとの接続を確認することです。

（注）

- テストアラートの生成は、実際の送信元からは生成されず、テスト目的でのみ生成されます。
- テストアラートは、少なくとも1つのパブリッシャにリンクされているアラートタイプに対して生成できます。

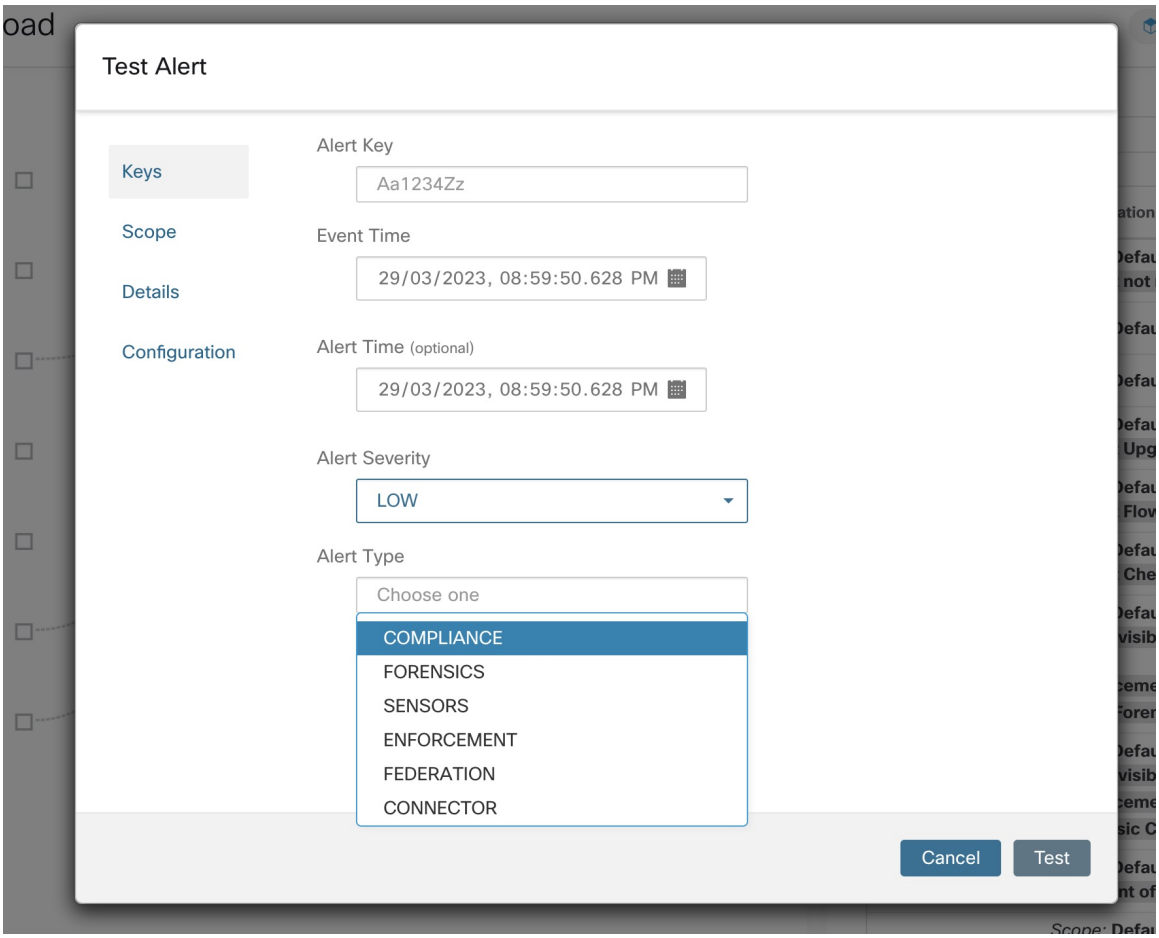
テストアラートを生成するには、次の手順に従います。

手順

ステップ 1 ナビゲーションウィンドウで、**[管理（Manage）]** > **[ワークロード（Workloads）]** > **[アラート設定（Alerts Config）]** をクリックします。

ステップ 2 テストアラートを設定するために、[テストアラート（Test Alert）] ボタンをクリックします。

図 *415*：テストアラートの設定ページ



[テストアラート（Test Alert）] ページが表示されます。

ステップ 3 [キー（Keys）] タブで、[アラートキー（Alert Key）] の値を入力し、[イベント時間（Event Time）]、[アラート時間（Alert Time）]、[アラートの重大度（Alert Severity）]、および[アラートタイプ（Alert Type）] の値を選択します。

ステップ 4 [範囲（Scope）] タブで、[範囲ID（Scope ID）] と [テナントID（Tenant ID）] の値が、現在の範囲に基づいて自動生成されます。

（注） [テナントID（Tenant ID）] が [テナントID VRF（Tenant ID VRF）] と同じである場合は、[テナントID VRF（Tenant ID VRF）] チェックボックスが自動的にオンになります。

ステップ 5 [詳細（Details）] タブで、[アラートテキスト（Alert Text）]、[イベントの注釈（Event Notes）]、[アラート詳細（Alert Details）]、および [アラート設定ID（Alert Configuration ID）] の値を入力します。

（注） [アラート詳細（Alert Details）] には、文字列値または JSON 形式のデータを指定できます。

JSON コンテンツのオプションは次のとおりです。

1. そのアラートタイプで想定されているフィールドを含める。
2. そのアラートタイプでデフォルトの json フィールドが想定されていない場合は、任意のサンプル JSON データ。

JSON の例：

```
{"alert_name ":"sample","alert_category":{"severity": "dummy"}}
```

ステップ 6 [設定（Configuration）] タブで、[個別アラート（Individual Alert）]、[アラート頻度（Alert Frequency）]、および [サマリーアラート頻度（Summary Alert Frequency）] の値を選択します。

個別アラートについては、ドロップダウンから [有効（ENABLE）] または [無効（DISABLE）] を選択します。

アラート頻度は、自動的に [個別（INDIVIDUAL）] の頻度に選択されます。

（注） 個別アラートのみがサポートされ、集約は考慮されません。

サマリーアラートは、自動的に [なし（NONE）] に選択されます。

ステップ 7 テストアラートを生成するために、[テスト（TEST）] をクリックします。

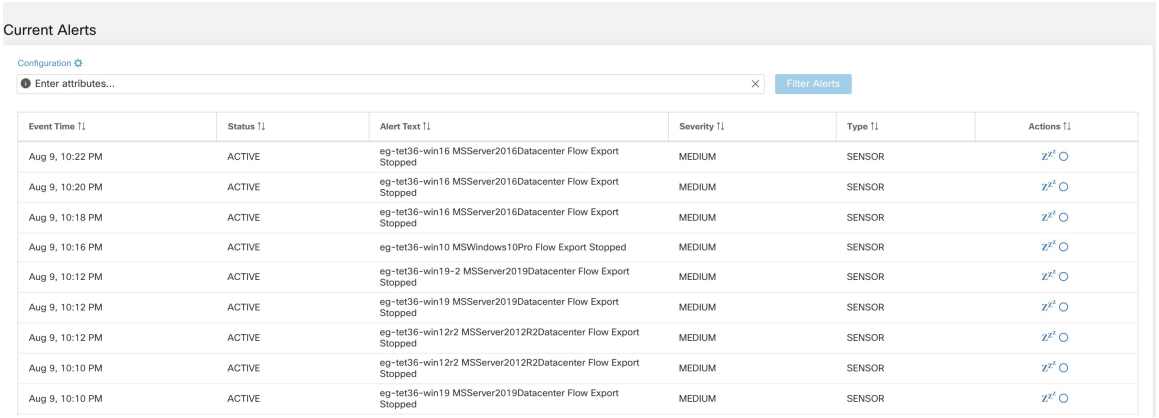
（注） テストアラートが生成され、設定されたパブリッシャに送信されます。

現在のアラート

[アラート（Alerts）] ページは次のように構成されています。アラートは、タイプ、ステータス（アクティブまたはスヌーズ）、および重大度（クリティカル、高、中、低）でフィルタリングできます。デフォルトでは、リストされているアラートはアクティブなアラートにフィルタリングされています（スヌーズおよびミュートされたアラートは、デフォルトでは表示されません）。

警告 [アラート（Alerts）] ページには、重大度の値が低（LOW）、中（MEDIUM）、または高（HIGH）のアラートのみが表示されます。重大度の値を問わず、すべてのアラートは必ず設定済みの Kafka ブローカーに送信されます。

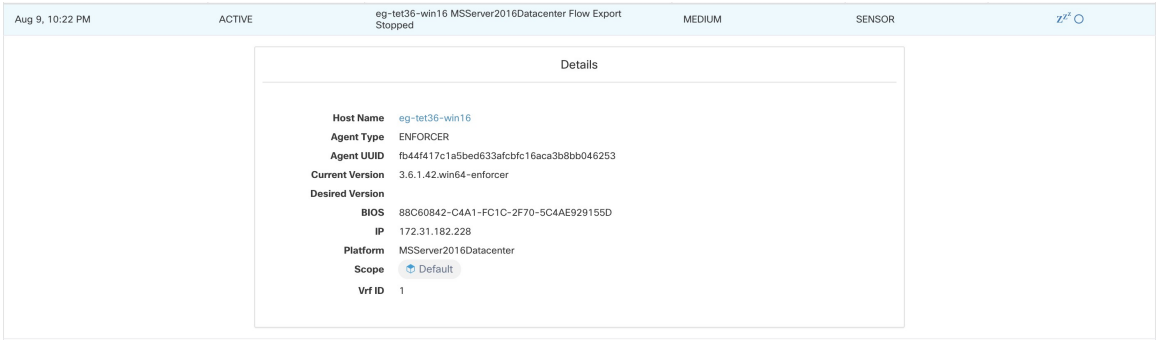
図 *416* : 現在のアラートリスト



アラート詳細の展開

特定のアラートの詳細が必要な場合は、アラートをクリックするだけで詳細情報が表示されます。

図 *417* : アラートの詳細



UI でのアラートの表示に関する注意事項

- UI に表示されるのは、ルート範囲当たり毎分 60 件のアラートのみです。アラートの量が増えると、UI に上述のサマリーアラートが表示されます。
- クリティカルアラート、次に重大度が高いアラート、次に重大度が中、最後に重大度が低いアラートの順に優先度が与えられます。
- ある時点で UI に表示されるアラートには最大数が設定されています。新しいアラートが届くと、古いアラートは削除されます。

「制限」を参照してください。

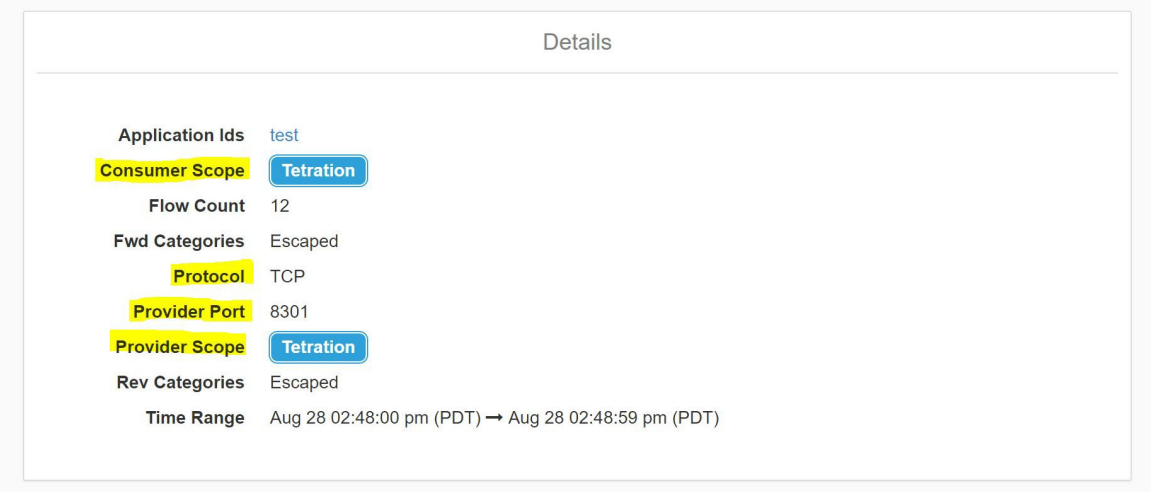
アラートのスヌーズ

（注） 現時点では、ユーザーアプリで作成されたアラートをスヌーズまたは無視（ミュート）することはできません。

Alerts アプリでは、同じ「タイプ」のアラートを、選択した時間だけスヌーズ（抑制）できます。「アラートのタイプ」の定義は、アラートが現在構成されているワークスペースに応じて異なることに注意してください。例としてコンプライアンスのアラートでは、「アラートのタイプ」は、コンシューマ範囲、プロバイダー範囲、プロトコル、プロバイダーポートの4つのタプルとして定義されます。

アラートのこれらのフィールドを表示するには、問題のアラートをクリックします。これだけで、アラートの詳細が表示されます。

図 *418*：アラートの詳細



アラートのスヌーズ

アラートをスヌーズするには、スヌーズする特定のアラートタイプの[アクション（Actions）]の下にあるスヌーズボタンをクリックし、期間を指定します。

図 *419*：アラートのスヌーズ

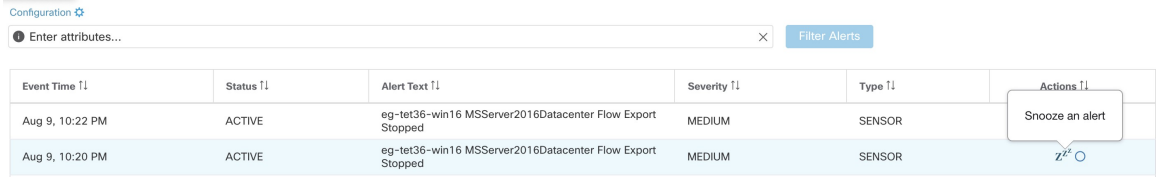
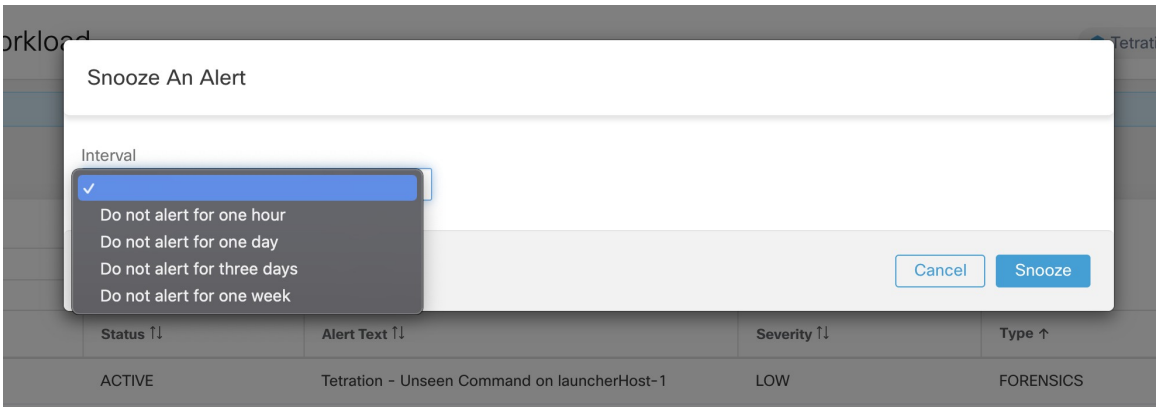


図 *420* : スヌーズ期間



図に示すように、アラートは4つの異なる期間（1時間、1日、3日、または1週間）でスヌーズできます。アラートのミュートは、本質的に永久にスヌーズするアクションであり、そのボタンも [アクション（Actions）] の下にあります。

図 *421* : ミュートリストに追加してアラートをミュートする

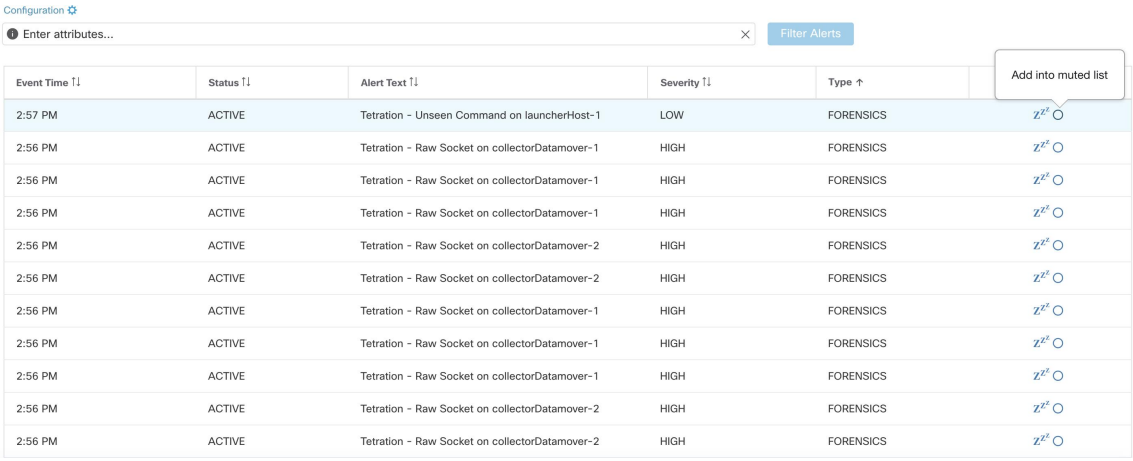
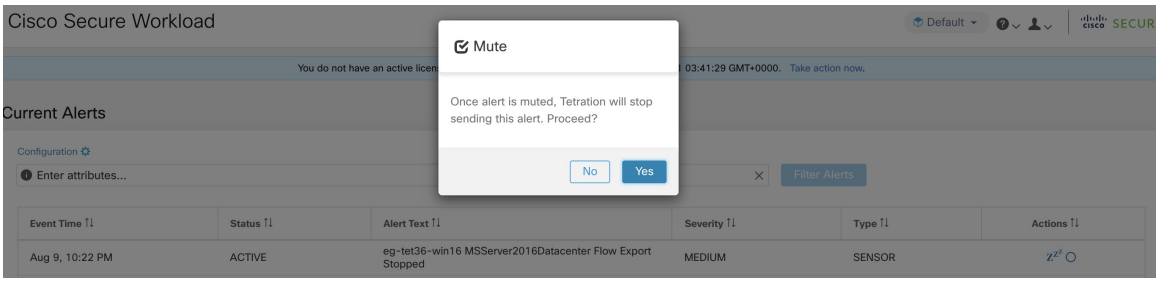


図 *422* : アラートのミュートの確認

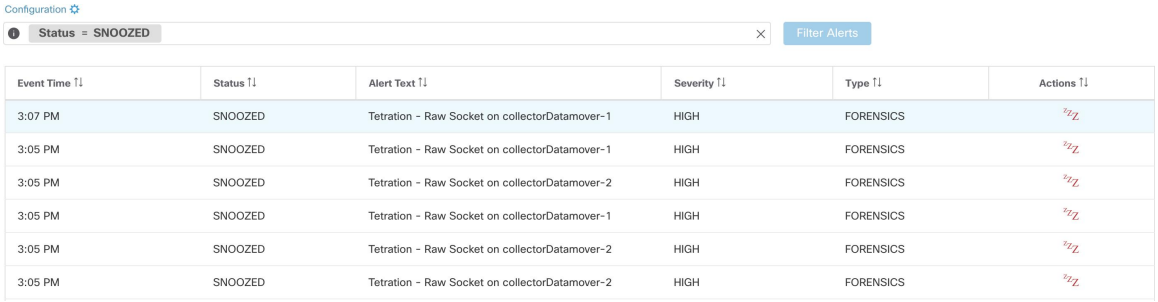


アラートが「ミュート」されている場合、アラートがミュートリストから削除されるまで、このタイプのアラートはユーザーに送信されません。

スヌーズまたはミュート状態の解除

以前にスヌーズされたアラートタイプのスヌーズを解除するには、最初にスヌーズされたアラートをフィルタで抽出して、スヌーズされたアラートのみが表示されるようにします。

図 *423* : スヌーズされたアラートフィルタ



次に、次の図のように [アクション（Actions）] で目的のアラートのスヌーズ解除ボタンをクリックし、アクションを確認します。

図 *424* : アラートのスヌーズ解除

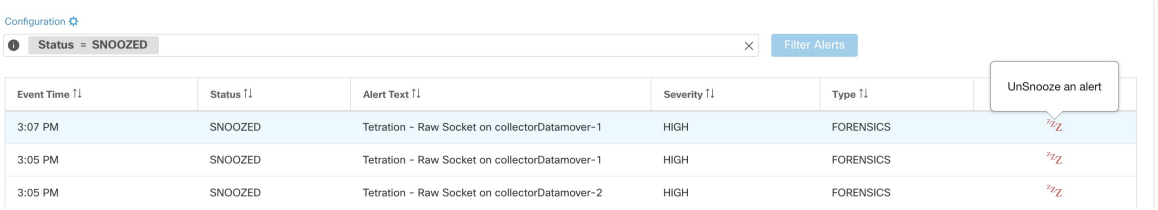
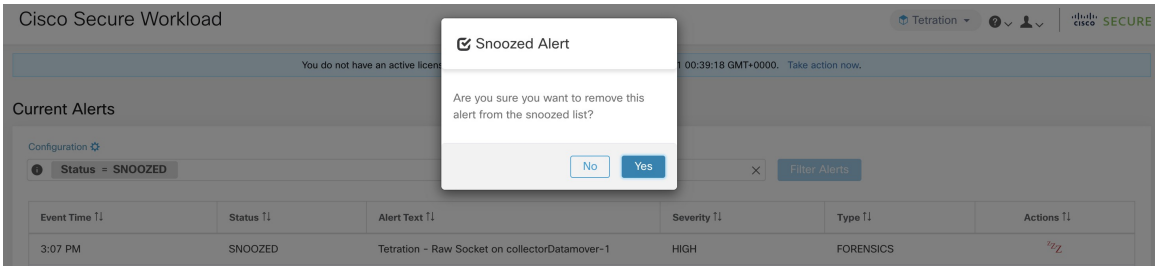


図 *425* : アラートのスヌーズ解除の確認



このプロセスは、ミュートされたアラート以外を除外するフィルタを使用して、ミュートリストからアラートを削除する場合も同じです。

図 *426* :「ミュートされた」アラートの選択とミュートリストからの削除

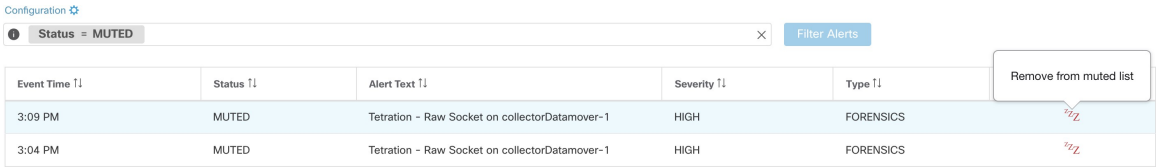
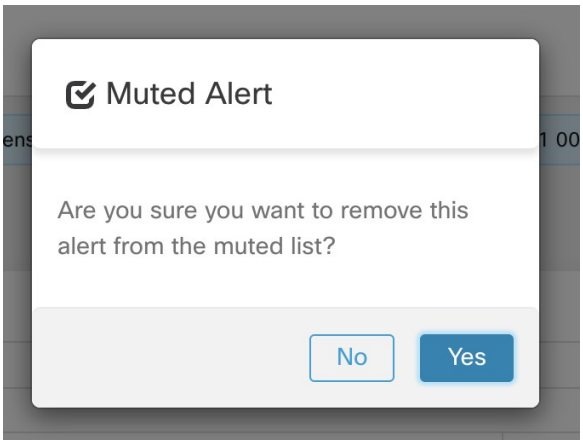


図 *427* : アラートのミュート解除の確認



アドミラルアラート

アドミラルは、以前のリリースの Bosun に代わる統合アラートシステムです。詳細については、後述する「アドミラルアラート」を参照してください。

アラート詳細

一般的なアラート構造

すべてのアラートは全体的な共通の構造に従いますが、アラートの詳細はアラートの種類ごとに異なります。

一般的な構造は次のとおりです。この構造は、Kafka データタップを通じて使用可能な json メッセージ構造に対応します。

フィールド	フォーマット	バージョン情報
root_scope_id	string	範囲の階層における最上位の範囲に対応する範囲 ID。
key_id	string	「類似の」アラートを決定するために使用される ID フィールド。同一の key_id をスヌーズできます。

フィールド	フォーマット	バージョン情報
type	string	アラートのタイプ。文字列値の固定セット：COMPLIANCE、USERAPP、FORENSICS、ENFORCEMENT、FABRIC、SENSOR、PLATFORM、FEDERATION、CONNECTOR
event_time	long	イベントがトリガーされたときのタイムスタンプ（またはイベントが範囲にまたがる場合は、範囲の開始点）。このタイムスタンプは、エポックミリ秒単位（UTC）です。
alert_time	long	アラートの送信が最初に試行されたときのタイムスタンプ。イベントの時間範囲の後になります。このタイムスタンプは、エポックミリ秒単位（UTC）です。
alert_text	string	アラートのタイトル。
alert_text_with_names	string	alert_text と同じ内容ですが、ID フィールドが対応する名前に置き換えられています。このフィールドは、すべてのアラートに存在するとは限りません。
severity	string	文字列値の固定セット：LOW、MEDIUM、HIGH、CRITICAL、IMMEDIATE_ACTION。このセットはアラートのシビラティ（重大度）です。一部のタイプのアラートでは、これらの値は設定可能です。
alert_notes	string	通常は設定しません。Kafka データタップを介して追加情報を渡すための特別なケースで存在する場合があります。

フィールド	フォーマット	バージョン情報
alert_conf_id	string	このアラートをトリガーしたアラート設定の ID。すべてのアラートに存在するとは限りません。
alert_details	string	構造化データ。String-i-fied json。このフィールドの正確な構造はアラートのタイプによって異なるため、特定のアラート タイプについては機能の詳細を参照してください。
alert_details_json	json	alert_details と同じ内容ですが、string-i-fied ではありません。コンプライアンスアラートにのみ存在し、Kafka を介してのみ使用できます。
tenant_Id	string	root_scope_id に対応する vrf を含む場合があります。または、デフォルト値として 0 が含まれている場合があります。まったく存在しない場合もあります。
alert_id	string	内部で生成された一時的な ID。無視して構いません。

フィールド	フォーマット	バージョン情報
root_scope_id	string	範囲の階層における最上位の範囲に対応する範囲 ID。
key_id	string	「類似の」アラートを決定するために使用される ID フィールド。同一の key_id をスヌーズできます。
type	string	アラートのタイプ。文字列値の固定セット：COMPLIANCE、USERAPP、FORENSICS、ENFORCEMENT、SENSOR、PLATFORM、FEDERATION、CONNECTOR。

フィールド	フォーマット	バージョン情報
event_time	long	イベントがトリガーされたときのタイムスタンプ（またはイベントが範囲にまたがる場合は、範囲の開始点）。このタイムスタンプは、エポックミリ秒単位（UTC）です。
alert_time	long	アラートの送信が最初に試行されたときのタイムスタンプ。イベントの時間範囲の後になります。このタイムスタンプは、エポックミリ秒単位（UTC）です。
alert_text	string	アラートのタイトル。
alert_text_with_names	string	alert_text と同じ内容ですが、ID フィールドが対応する名前に置き換えられています。このフィールドは、すべてのアラートに存在するとは限りません。
severity	string	文字列値の固定セット：LOW、MEDIUM、HIGH、CRITICAL、IMMEDIATE_ACTION。このセットはアラートのシビラティ（重大度）です。一部のタイプのアラートでは、これらの値は設定可能です。
alert_notes	string	通常は設定しません。Kafka データタップを介して追加情報を渡すための特別なケースで存在する場合があります。
alert_conf_id	string	このアラートをトリガーしたアラート設定の ID。すべてのアラートに存在するとは限りません。

フィールド	フォーマット	バージョン情報
alert_details	string	構造化データ。String-i-fied json。このフィールドの正確な構造はアラートのタイプによって異なるため、特定のアラート タイプについては機能の詳細を参照してください。
alert_details_json	json	alert_details と同じ内容ですが、string-i-fied ではありません。コンプライアンスアラートにのみ存在し、Kafka を介してのみ使用できます。
tenant_Id	string	root_scope_id に対応する vrf を含む場合があります。または、デフォルト値として 0 が含まれている場合があります。まったく存在しない場合もあります。
alert_id	string	内部で生成された一時的な ID。無視して構いません。

alert_details 内のフィールドは、アラートのタイプによって異なります。説明とフィールドのリストについては、各機能のセクションを参照してください。

- コンプライアンス：lab-compliance-alert-details
- 近接：アラート詳細
- フォレンジック：外部との統合およびフォレンジックイベントに表示されるフィールド
- センサー：センサーアラートの詳細
- 適用：適用アラートの詳細
- コネクタ：アラート詳細

- コンプライアンス：lab-compliance-alert-details
- フォレンジック：外部との統合およびフォレンジックイベントに表示されるフィールド
- センサー：センサーアラートの詳細
- 適用：適用アラートの詳細
- コネクタ：アラート詳細

オンプレミスクラスタにおける追加のアラートタイプ

- ファブリック：fabric-alert-details
- フェデレーション：federation-alert-details
- プラットフォーム：アラート詳細

- フェデレーション：federation-alert-details
- プラットフォーム：アラート詳細

通知機能別の全般アラート形式

アラート形式は、通知タイプごとに異なります。以下に、さまざまな通知タイプでアラートがどのように表示されるかの例を示します。

Kafka（DataTap）

Kafka（DataTap）メッセージは JSON 形式です。以下に例を示します。その他の例については、前述の alert_details を参照してください。


```
  {
  "severity": "LOW",
  "tenant_id": 0,
  "alert_time": 1595207103337,
  "alert_text": "Lookout Annotated Flows contains TA_zeus for
<scope_id:5efcfdf5497d4f474f1707c2>",
  "key_id": "0a4a4208-f721-398c-b61c-c07af3be9413",
  "alert_id":
"/Alerts/5efcfdf5497d4f474f1707c2/DataSource{location_type='TETRATION_PARQUET',
location_name='lookout_annotation', location_grain='HOURLY',
root_scope_id='5efcfdf5497d4f474f1707c2'}/bd33f37af32a5ce71e888f95ccfe845305e61a12a7829ca5f2d72bf96237d403",

  "alert_text_with_names": "Lookout Annotated Flows contains TA_zeus for Scope Default",

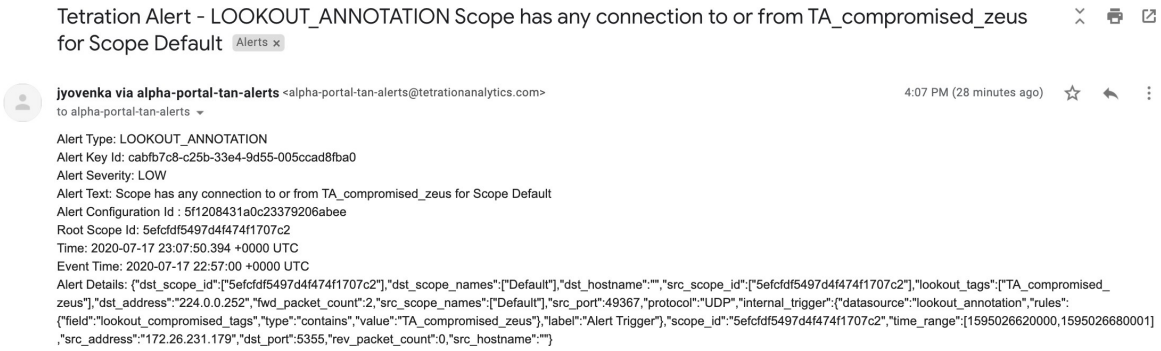
  "root_scope_id": "5efcfdf5497d4f474f1707c2",
  "alert_conf_id": "5f10c7141a0c236b78148da1",
  "type": "LOOKOUT_ANNOTATION",
  "event_time": 1595204760000,
  "alert_details":
'{\"type\":\"Lookout Annotated Flows\",\"trigger\":\"{\\\"scope_id\\\":\\\"5efcfdf5497d4f474f1707c2\\\",\\\"time_range\\\":[1595204760000,1595204820001],\\\"src_address\\\":\\\"172.26.230.124\\\",\\\"dst_port\\\":137,\\\"ev_packet_count\\\":0,\\\"src_hostname\\\":\\\"\\\"}"

  }
```


E メール

E メールアラートの設定に関する情報：Email Connector

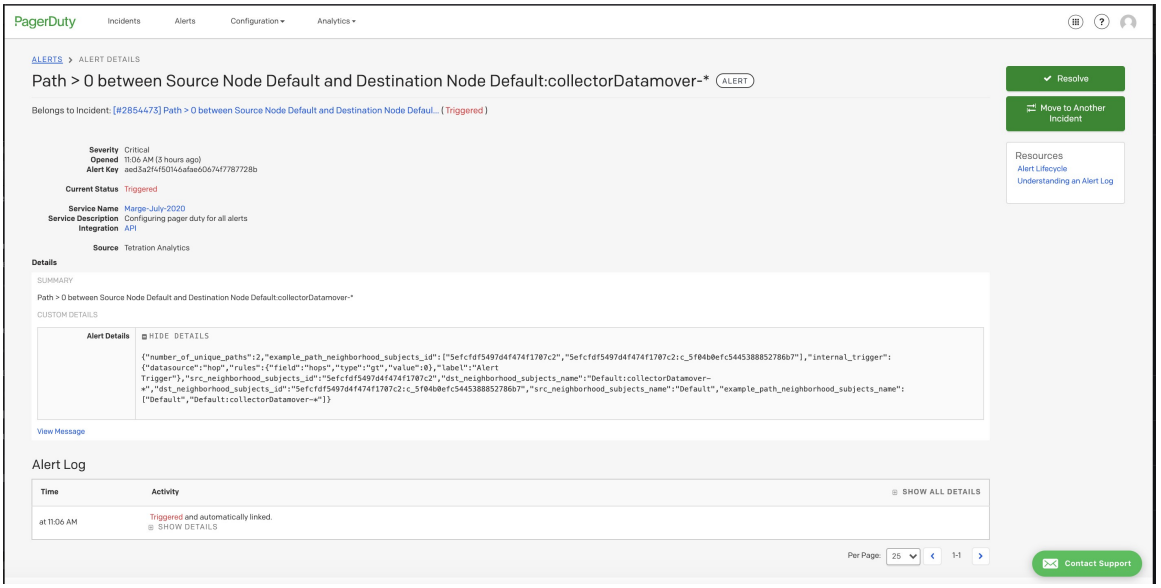
図 *428 : E* メール送信が設定されている場合の *Cisco Secure Workload* アラートの例



PagerDuty

PagerDuty アラートの設定に関する情報：PagerDuty Connector

図 *429 : PagerDuty* における *Secure Workload* アラートの例



PagerDuty に送信されたアラートは、key_id に基づく同じアラートの再トリガーと見なされます。

シビラリティ（重大度）は、次のように PagerDuty のシビラリティ（重大度）にマッピングされます。

Cisco Secure Workload のシビラリティ（重大度）	PagerDuty のシビラティ（重大度）
IMMEDIATE_ACTION	critical
CRITICAL	critical
HIGH	error

Cisco Secure Workload のシビラリティ（重大度）	PagerDuty のシビラティ（重大度）
MEDIUM	warning
LOW	info

Syslog

Syslogアラートの設定およびシビラリティ（重大度）のマッピングの調整については、「Syslog Connector」を参照してください。

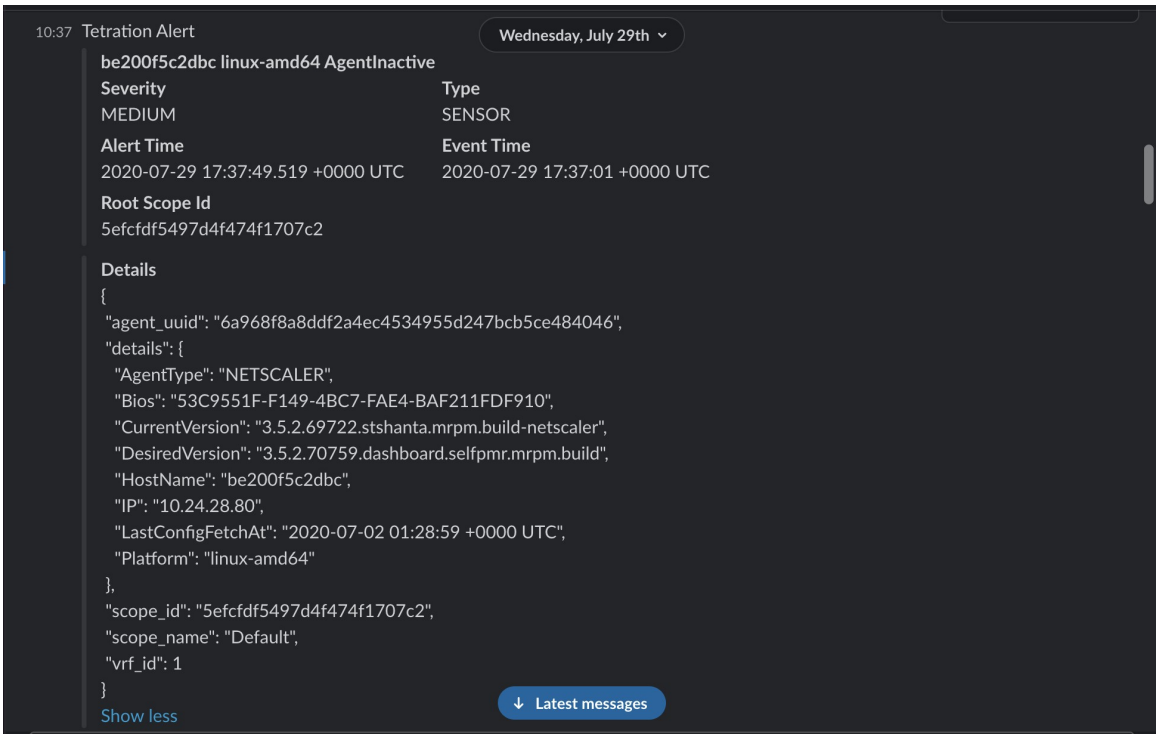
図 *430 : Syslog* に送信された *Secure Workload* アラートの例


```
Aug  2 18:45:21 tan-5f035bae1a0c231d5880d7f8-tac-demo-data-ingest Tetration Alert[26841]: [DEBUG] {"keyId":"3ee0d8b7-bc81-3427-9e84-6b9f8fedb98c","eventTime":"159639
3720000","alertTime":"1596393968822","alertText":"Enforcement Annotated Flows contains escaped for \u003capplication_id:5f04b0b9755f024d4e36a279\u003e","severity":"L
OW","tenantId":"0","type":"COMPLIANCE","alertDetails":"{\"consumer_scope_ids\":[\"5efcfdf5497d4f474f1707c2\"],\"consumer_scope_names\":[\"Default\"],\"provider_scope
_names\":[\"Default\"],\"provider_port\":53,\"application_id\":\"5f04b0b9755f024d4e36a279\",\"constituent_flows\":[{\"consumer_port\":37367,\"protocol\":\"UDP\",\"co
nsumer_address\":\"172.31.163.139\",\"provider_address\":\"171.70.168.183\",\"provider_port\":53},{\"consumer_port\":39652,\"protocol\":\"UDP\",\"consumer_address\":
\"172.31.163.137\",\"provider_address\":\"171.70.168.183\",\"provider_port\":53},{\"consumer_port\":63811,\"protocol\":\"UDP\",\"consumer_address\":\"172.31.163.136\
",\"provider_address\":\"171.70.168.183\",\"provider_port\":53},{\"consumer_port\":57418,\"protocol\":\"UDP\",\"consumer_address\":\"172.31.163.138\",\"provider_addr
ess\":\"173.36.131.10\",\"provider_port\":53},{\"consumer_port\":12599,\"protocol\":\"UDP\",\"consumer_address\":\"172.31.163.141\",\"provider_address\":\"173.36.131
.10\",\"provider_port\":53},{\"consumer_port\":7385,\"protocol\":\"UDP\",\"consumer_address\":\"172.31.163.140\",\"provider_address\":\"173.36.131.10\",\"provider_po
rt\":53}],\"escaped_count\":6,\"provider_scope_ids\":[\"5efcfdf5497d4f474f1707c2\"],\"policy_type\":\"ENFORCED_POLICY\",\"protocol\":\"UDP\",\"internal_trigger\":{\"
datasource\":\"compliance\",\"rules\":{\"field\":\"policy_violations\",\"type\":\"contains\",\"value\":\"escaped\"},\"label\":\"Alert Trigger\"},\"time_range\":[1596
393720000,1596393779999],\"policy_category\":[\"ESCAPED\"]}","rootScopeId":"5efcfdf5497d4f474f1707c2","alertConfId":"5f15cca71a0c231ebd66ca3b","alertTextWithNames":"
Enforcement Annotated Flows contains escaped for Enforced Application j1"}
Aug  2 18:45:21 tan-5f035bae1a0c231d5880d7f8-tac-demo-data-ingest Tetration Alert[26841]: [DEBUG] {"keyId":"8f0cfcb5-f8c1-3130-a069-3721b7d50159","eventTime":"159639
3720000","alertTime":"1596393968822","alertText":"Enforcement Annotated Flows contains escaped for \u003capplication_id:5f04b0b9755f024d4e36a279\u003e","severity":"L
OW","tenantId":"0","type":"COMPLIANCE","alertDetails":"{\"consumer_scope_ids\":[\"5efcfdf5497d4f474f1707c2\"],\"consumer_scope_names\":[\"Default\"],\"provider_scope
_names\":[\"Default\"],\"provider_port\":5660,\"application_id\":\"5f04b0b9755f024d4e36a279\",\"constituent_flows\":[{\"consumer_port\":17131,\"protocol\":\"TCP\",\"
consumer_address\":\"172.26.231.193\",\"provider_address\":\"172.31.163.140\",\"provider_port\":5660}],\"escaped_count\":1,\"provider_scope_ids\":[\"5efcfdf5497d4f47
4f1707c2\"],\"policy_type\":\"ENFORCED_POLICY\",\"protocol\":\"TCP\",\"internal_trigger\":{\"datasource\":\"compliance\",\"rules\":{\"field\":\"policy_violations\",\
"type\":\"contains\",\"value\":\"escaped\"},\"label\":\"Alert Trigger\"},\"time_range\":[1596393720000,1596393779999],\"policy_category\":[\"ESCAPED\"]}","rootScopeI
d":"5efcfdf5497d4f474f1707c2","alertConfId":"5f15cca71a0c231ebd66ca3b","alertTextWithNames":"Enforcement Annotated Flows contains escaped for Enforced Application j1
"}
Aug  2 18:45:21 tan-5f035bae1a0c231d5880d7f8-tac-demo-data-ingest Tetration Alert[26841]: [DEBUG] {"keyId":"1ef4a974-be89-31de-abe9-dc71cb0170ad","eventTime":"159639
3720000","alertTime":"1596393968822","alertText":"Enforcement Annotated Flows contains escaped for \u003capplication_id:5f04b0b9755f024d4e36a279\u003e","severity":"L
OW","tenantId":"0","type":"COMPLIANCE","alertDetails":"{\"consumer_scope_ids\":[\"5efcfdf5497d4f474f1707c2\"],\"consumer_scope_names\":[\"Default\"],\"provider_scope
_names\":[\"Default\"],\"provider_port\":443,\"application_id\":\"5f04b0b9755f024d4e36a279\",\"constituent_flows\":[{\"consumer_port\":17792,\"protocol\":\"TCP\",\"c
onsumer_address\":\"172.26.231.193\",\"provider_address\":\"172.31.163.133\",\"provider_port\":443}],\"escaped_count\":1,\"provider_scope_ids\":[\"5efcfdf5497d4f474f
1707c2\"],\"policy_type\":\"ENFORCED_POLICY\",\"protocol\":\"TCP\",\"internal_trigger\":{\"datasource\":\"compliance\",\"rules\":{\"field\":\"policy_violations\",\"t
ype\":\"contains\",\"value\":\"escaped\"},\"label\":\"Alert Trigger\"},\"time_range\":[1596393720000,1596393779999],\"policy_category\":[\"ESCAPED\"]}","rootScopeId"
:"5efcfdf5497d4f474f1707c2","alertConfId":"5f15cca71a0c231ebd66ca3b","alertTextWithNames":"Enforcement Annotated Flows contains escaped for Enforced Application j1"}
```


Slack

Slack アラートの構成に関する情報：Slack コネクタ

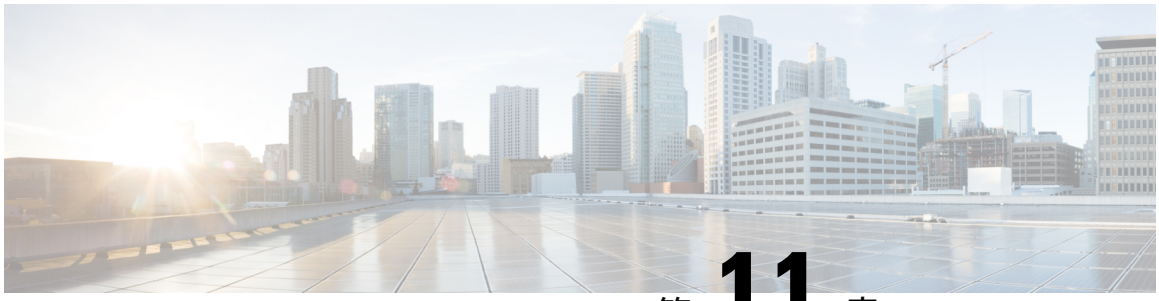
図 *431 : Slack* チャネルに送信される *Secure Workload* アラートの例



Kinesis

Kinesis アラートの構成に関する情報：Kinesis コネクタ

Kinesis アラートは、Kafka アラートに似ています。どちらもメッセージキューであるためです。



第 11 章

メンテナンス

表示されるメンテナンスオプションは、ユーザーロールによって異なります。

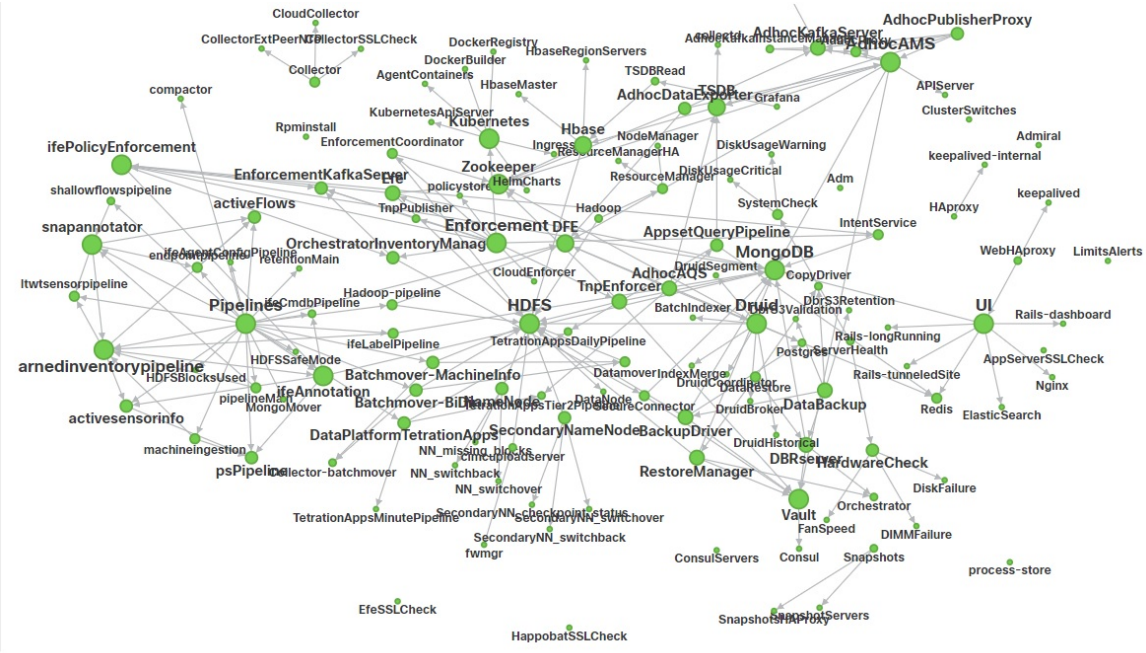
- サービスのステータス （813 ページ）
- アドミラルアラート （814 ページ）
- クラスタのステータス （824 ページ）
- データのバックアップと復元 （830 ページ）
- VM の情報 （852 ページ）
- クラスタのアップグレード （852 ページ）
- スナップショット （862 ページ）
- Explore / スナップショットのエンドポイントの概要 （871 ページ）
- サーバーのメンテナンス （887 ページ）
- ディスク メンテナンス （895 ページ）
- クラスタのメンテナンス：クラスタのシャットダウンと再起動 （909 ページ）
- [データタップ管理者（Data Tap Admin）]：データのタップ （912 ページ）

サービスのステータス

左側のナビゲーションバーの [トラブルシュート（Troubleshoot）] メニューの下にある [サービスステータス（Service Status）] ページには、Cisco Secure Workload クラスタで使用されている全サービスの正常性と、サービスの依存関係が表示されます。

グラフビューにはサービスの正常性が表示されます。グラフの各ノードにはサービスの正常性が表示され、エッジは他のサービスへの依存関係を表します。異常なサービスは、サービスが利用できない場合は赤色、サービスが低下しているが利用可能な場合はオレンジ色で示されます。緑色のノードは、サービスが正常であることを示します。ノードに関する詳細なデバッグ情報を確認する場合は、[すべて展開（Expand All）] ボタンがあるツリービューを使用して、依存関係ツリー内のすべての子ノードを表示します。「ダウン」はサービスが機能していないことを示し、「異常」はサービスが完全には機能していないことを示します。

図 *432 : [*サービスステータス（***Service Status***）*]* ページ



アドミラルアラート

アドミラルは、統合されたアラートシステムです。サービスのステータスによって報告されたサービスの正常性に基づいてアラートを処理します。そのため、統一された方法でユーザーはサービス/クラスタの健全性を判断することができます。サービスステータスは、サービスの現在（特定の時点）の状態を示します。サービスステータスが赤色と報告された場合、サービスは停止していると見なされ、それ以外の場合は稼働していると見なされます。稼働時間は、サービスが稼働していると報告される時間です。アドミラルは、一定期間にわたるサービスステータスによって報告されたサービスの正常性を評価し、サービスの稼働時間の割合が特定のしきい値を下回った場合にアラートを生成します。一定期間にわたってこの評価を行うことで、誤検知を減らし、本当のサービス停止時にのみ警告を発することが保証されます。

サービスによってアラートのニーズは異なるため、この割合と時間間隔は、サービスごとに異なる方法で固定されています。

顧客は、アドミラル通知を使用して、これらのイベントの通知を受けることができます。これらのイベントは、プラットフォームタイプの **[調査（Investigate）]**>**[アラート（Alerts）]** ページにも表示されます。

（注） アドミラルアラートは、選択したサービスのサブセットのみに関連付けられます。サービスが上記のサブセットに含まれていない場合、サービスがダウンしてもアドミラルアラートは発生しません。このサービスのサブセットで設定されているアドミラルアラートとそのアラートしきい値の割合と時間間隔は固定であり、ユーザーは設定できません。

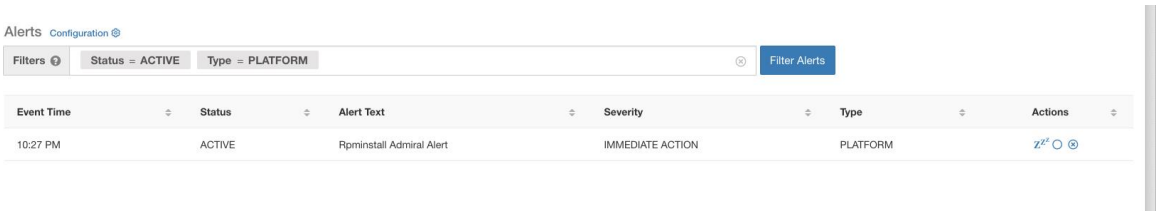
次のセクションでは、アドミラルアラートと通知について詳しく説明します。

アドミラルアラートのライフサイクル

アドミラルは、サービスステータスでサービスの稼働時間をチェックします。この稼働時間があらかじめ設定されたアラート用のしきい値を下回ると、アラートが発生します。

たとえば、RPminstall は、展開、アップグレード、パッチなどの際に rpm をインストールするために使用されるサービスです。1 時間以上の稼働時間が 80% 未満の場合、アドミラルアラートを生成するように設定されています。Rpminstall サービスが上で指定されたしきい値よりも長い期間ダウンした場合、RPminstall のアドミラルアラートが生成され、ステータスが ACTIVE になります。

図 *433*：アクティブなアドミラルアラート



サービスが回復すると、稼働時間の割合が増加し始めます。稼働時間がしきい値を超えると、アラートは自動的にクローズし、ステータスは CLOSED に移行します。上記の Rpminstall の例では、稼働時間が 1 時間で 80% を超えると、Rpminstall アドミラルアラートは自動的にクローズします。

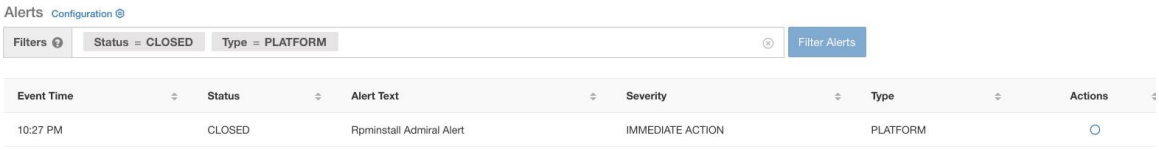
（注） アラートのクローズにより、サービスは常に正常に戻るのが遅れます。これは、アドミラルが一定期間サービス正常性を監視するためです。上記の例では、RPminstall アラートのしきい値が 1 時間の稼働時間の 80% に設定されているため、アラートがクローズするまでに少なくとも 48 分間（1 時間の 80%）稼働している必要があります。

アラートをクローズするために必要なユーザーのアクションはありません。アクティブなアドミラルアラートは、注意が必要な現在の根本的な問題を示すようになります。

（注） アラートがクローズしても、専用の通知は生成されません。

アラートが CLOSED に移動すると、ACTIVE アラートの下に表示されなくなります。クローズされたアラートは、次に示すように、フィルタの Status=CLOSED を使用して、UI に引き続き表示されます。

図 *434* : サービス回復時に自動的にクローズするアドミラルアラート



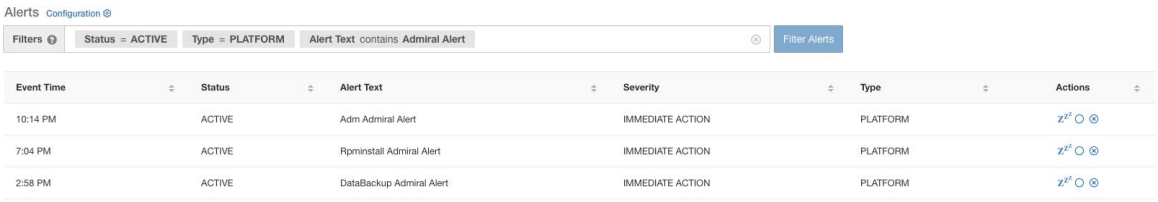
アドミラルアラートには次の 2 種類があります。

1. 個別のアドミラルアラート
2. サマリーのアドミラルアラート

個別のアドミラルアラート

上記で説明したアラート、個々のサービスに対して発生したアラートは、このカテゴリに分類されます。これらのアラートのアラートテキストには常に <Service Name> Admiral Alert が含まれています。これにより、個々のアラートをサービスまたは「Admiral Alert」サフィックスで簡単にフィルタリングできます。

図 *435* : 個別のアドミラルアラートのアラートテキストフィルタ



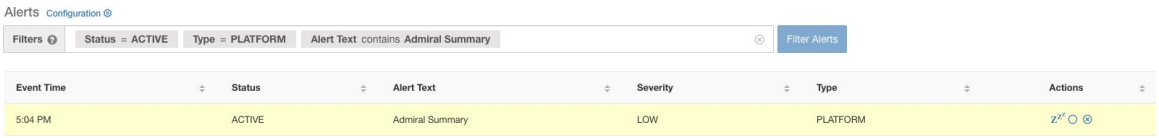
このサービスのその他の属性については、_admiral_indiv_details-label で説明しています。

サマリーのアドミラルアラート

アドミラルは、UTC の午前 0 時に毎日サマリーアラートを生成します。サマリーアラートには、現在アクティブなアラートと、過去1日以内にクローズされたすべてのアラートのリストが含まれているため、ユーザーは、アドミラルによって報告された全体的なクラスタの正常性を1か所で確認できます。これは、専用の通知を生成しないクローズされたアラートを表示する場合にも役立ちます。クラスタが正常で、過去1日以内にクローズされたアラートがない場合、その日のサマリー通知は生成されません。これは、不要な通知とノイズを減らすために行われます。

この場合のアラートテキストは常に「アドミラルサマリー」なので、以下に示すように、サマリーアラートを簡単にフィルタ処理できます。

図 *436 :* アドミラル サマリー テキスト フィルタ



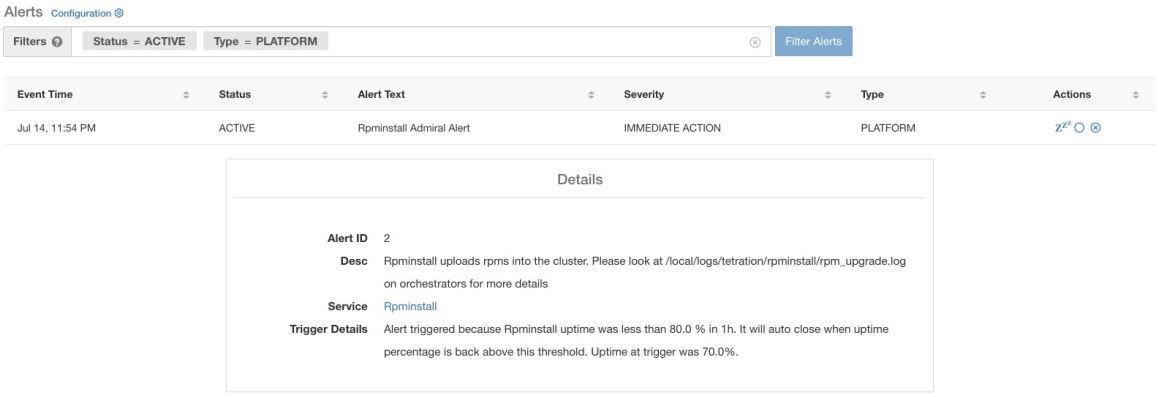
このサービスのその他の属性については、_admiral_summary_dets-label を参照してください。

アラート詳細

個別のアラート

個別のアドミラルアラートのアラートをクリックすると、アラートが展開され、アラートのデバッグと分析に役立つフィールドが表示されます。

図 *437 :* アラート詳細



以下の表で各フィールドについて説明します。

フィールド	バージョン情報
アラートID（Alert ID）	各アラートには、アラート ID と呼ばれる一意の ID があります。この ID は、特定のサービスダウン発生を一意に把握するのに役立ちます。前述のように、アラートによってレポートされているサービスの基本的な稼働時間が正常になると、アラートは自動的に閉じます。その後、同じサービスが再びダウンすると、別のアラート ID を持つ新しいアラートが生成されます。このように、アラート ID は、発生したアラートの各インシデントを一意に把握するのに役立ちます。
Desc	説明フィールドには、アラートの原因となっているサービスの問題についての追加情報が含まれています。

フィールド	バージョン情報
サービス	このフィールドには、ユーザーがサービスの現在のステータスを確認できるサービスステータスページへのリンクが含まれています。ユーザーは、サービスステータスページでサービスがダウンとマークされている理由の詳細を把握することもできます。
トリガーの詳細情報	このフィールドには、サービスのトリガーしきい値に関する詳細情報が含まれます。ユーザーは、これらのしきい値を確認することで、基本的なサービスが復旧した後にアラートが閉じるタイミングを把握できます。例：Rpminstallのしきい値は、1時間で80%の稼働時間と記されています。したがって、アラートが自動的に閉じる前に、Rpminstallサービスは少なくとも48分間（1時間の80%）稼働している必要があります。ここには、アラートが発生した時点でサービスに表示された稼働時間の値も示されています。

JSON Kafka出力の例は次のとおりです。

```
  {
  "severity": "IMMEDIATE_ACTION",
  "tenant_id": 0,
  "alert_time": 1595630519423,
  "alert_text": "Rpminstall Admiral Alert",
  "key_id": "ADMIRAL_ALERT_5",
  "alert_id": "/Alerts/5efcfdf5497d4f474f1707c2/DataSource{location_type='TETRATION',
location_name='platform', location_grain='MIN',
root_scope_id='5efcfdf5497d4f474f1707c2'}/66eb975f5f987fe9eaefa81cee757c8b6dac5facc26554182d8112a98b35c4ab",

  "root_scope_id": "5efcfdf5497d4f474f1707c2",
  "type": "PLATFORM",
  "event_time": 1595630511858,
  "alert_details": "{\"Alert ID\":5,\"Service\":\"Rpminstall\",\"Desc\":\"Rpminstall
uploads rpms into the cluster. Please look at
/local/logs/tetration/rpminstall/rpm_upgrade.log on orchestrators for more
details\",\"Trigger Details\":\"Alert triggered because Rpminstall uptime was less than
 80.0 % in 1h. It will auto close when uptime percentage is back above this threshold.
Uptime at trigger was 65.0%. \"}"
  }
```

個別のアラートはすべて、上記の形式に従います。アドミラルモニタリングの対象となる（サービスステータスからの）サービスのリストを表に示します。

サービス	トリガー条件	重大度
KubernetesApiServer	過去15分間でサービスの稼働時間が90%を下回っている。	即時対応（IMMEDIATE ACTION）

サービス	トリガー条件	重大度
Adm	過去1時間でサービスの稼働時間が90%を下回っている。	即時対応（IMMEDIATE ACTION）
DataBackup	過去6時間でサービスの稼働時間が90%を下回っている。	即時対応（IMMEDIATE ACTION）
DiskUsageCritical	過去1時間でサービスの稼働時間が80%を下回っている。	即時対応（IMMEDIATE ACTION）
RebootRequired	過去1時間でサービスの稼働時間が90%を下回っている。	即時対応（IMMEDIATE ACTION）
Rpminstall	過去1時間でサービスの稼働時間が80%を下回っている。	即時対応（IMMEDIATE ACTION）
SecondaryNN_checkpoint_status	過去1時間でサービスの稼働時間が90%を下回っている。	即時対応（IMMEDIATE ACTION）

8RU/39 RU 物理クラスタの場合、次のサービスが追加でモニタリングされます。

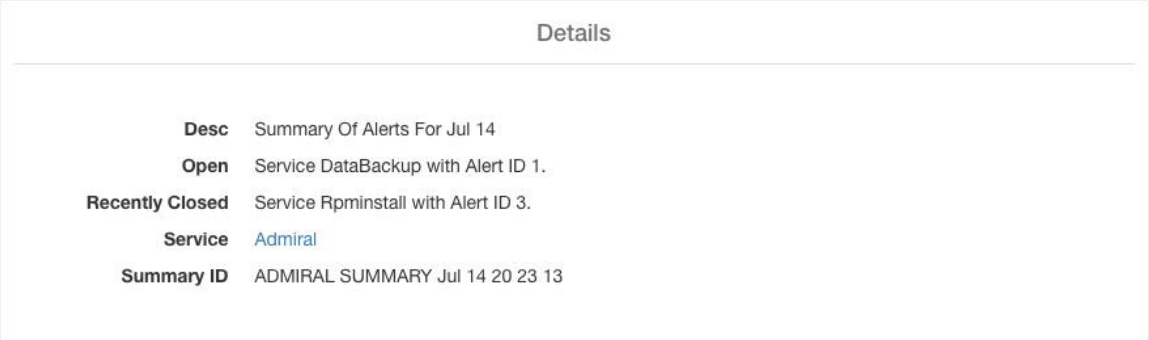
サービス	トリガー条件	重大度
DIMMFailure	過去 1 時間でサービスの稼働時間が 80% を下回っている。	即時対応（IMMEDIATE ACTION）
DiskFailure	過去 1 時間でサービスの稼働時間が 80% を下回っている。	即時対応（IMMEDIATE ACTION）
FanSpeed	過去 1 時間でサービスの稼働時間が 80% を下回っている。	即時対応（IMMEDIATE ACTION）
ClusterSwitches	過去 1 時間でサービスの稼働時間が 80% を下回っている。	即時対応（IMMEDIATE ACTION）

（注） Admiral は、サービスステータスによって生成された処理メトリックに依存してアラートを生成します。メトリックの取得が長期間不可能な場合（たとえば、サービスステータスが停止している場合）、アラート（TSDBOracleConnectivity）が発生し、クラスタでサービスベースのアラート処理がオフになっていることを通知します。

サマリーアラート

サマリーアラートは本質的に情報提供であり、優先順位は常に LOW に設定されます。アドミラルサマリーアラートをクリックすると、アドミラルアラートに関する概要情報を含む複数のフィールドが展開されて表示されます。

図 *438 :* アドミラルサマリーアラートの詳細



フィールド	バージョン情報
Desc	説明フィールドには、日次概要の日付が含まれています。
オープン（Open）	オープンアラートは、概要が生成された時点でアクティブだったアラートを示しています 。
[最近閉じたアラート（Recently Closed）]	このフィールドには、過去24時間以内、つまり概要が生成された日に閉じたアラートが表示されます。各アラートのIDも含まれます。アラートは自動的に閉じるため、特定のサービスがダウンしてアラートが作成された後、正常になり、アラートが自動的に閉じる場合があります。アラートが閉じるケースが 1 日に複数回発生した場合、各インシデントとその固有のアラートIDが一覧表示されます。ただし、アラートが閉じる前に各サービスがしきい値時間の間稼働状態になっている必要があることを考えると、こうした状況が頻繁に発生することは想定されていません。ユーザーは、Status＝CLOSEDでフィルタリングして、各インシデントに関する詳細情報を取得できます。

フィールド	バージョン情報
サービス	サービスを処理し、日次概要を生成する Admiral のサービスステータスリンク。
[サマリーID（Summary ID）]	サマリーアラートの ID。

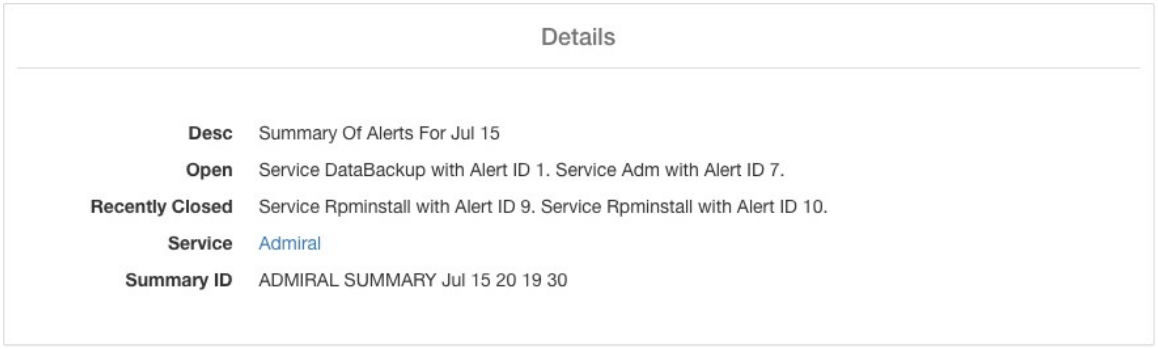
JSON Kafka 出力の例は次のとおりです。


```
  {
    "severity": "LOW",
    "tenant_id": 0,
    "alert_time": 1595721914808,
    "alert_text": "Admiral Summary",
    "key_id": "ADMIRAL_SUMMARY_Jul-26-20-00-04",
    "alert_id": "/Alerts/5efcfdf5497d4f474f1707c2/DataSource{location_type='TETRATION',
 location_name='platform', location_grain='MIN',
root_scope_id='5efcfdf5497d4f474f1707c2'}/e95da4521012a4789048f72a791fb58ab233bbff63e6cbc421525d4272d469aa",

    "root_scope_id": "5efcfdf5497d4f474f1707c2",
    "type": "PLATFORM",
    "event_time": 1595721856303,
    "alert_details": "{\"Desc\":\"Summary of alerts for Jul-26\",\"Recently
Closed\":\"None\",\"Open\":\" Service Rpminstall with Alert ID
5.\",\"Service\":\"Admiral\",\"Summary ID\":\"ADMIRAL_SUMMARY_Jul-26-20-00-04\"}"
  }
```


1 日に複数のアラートを発生させるサービスを含むサマリーアラートの例を以下に示します。

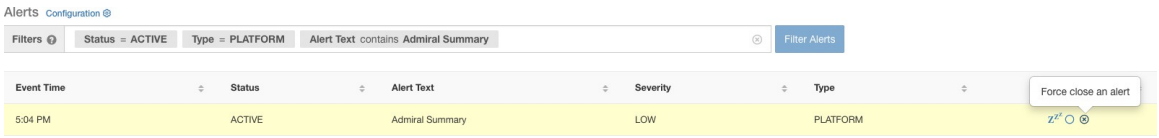
図 *439 :* 複数のアラート



ユーザのアクション

アドミラルアラートはアラートごとに1回だけ個別の通知を生成するため、特定のアラートを含めたり除外したり、スヌーズしたりする必要はありません。上述のとおり、しきい値である稼働時間の間サービスが正常に動作すると、アラートが自動的に閉じます。アラートを強制的に閉じるための強制終了オプションがあります。個々のアラートは自動的に閉じるため、通常、このオプションの使用は、UI からサマリーアラートを削除する場合に限る必要があります。

図 *440 :* アラートの強制終了

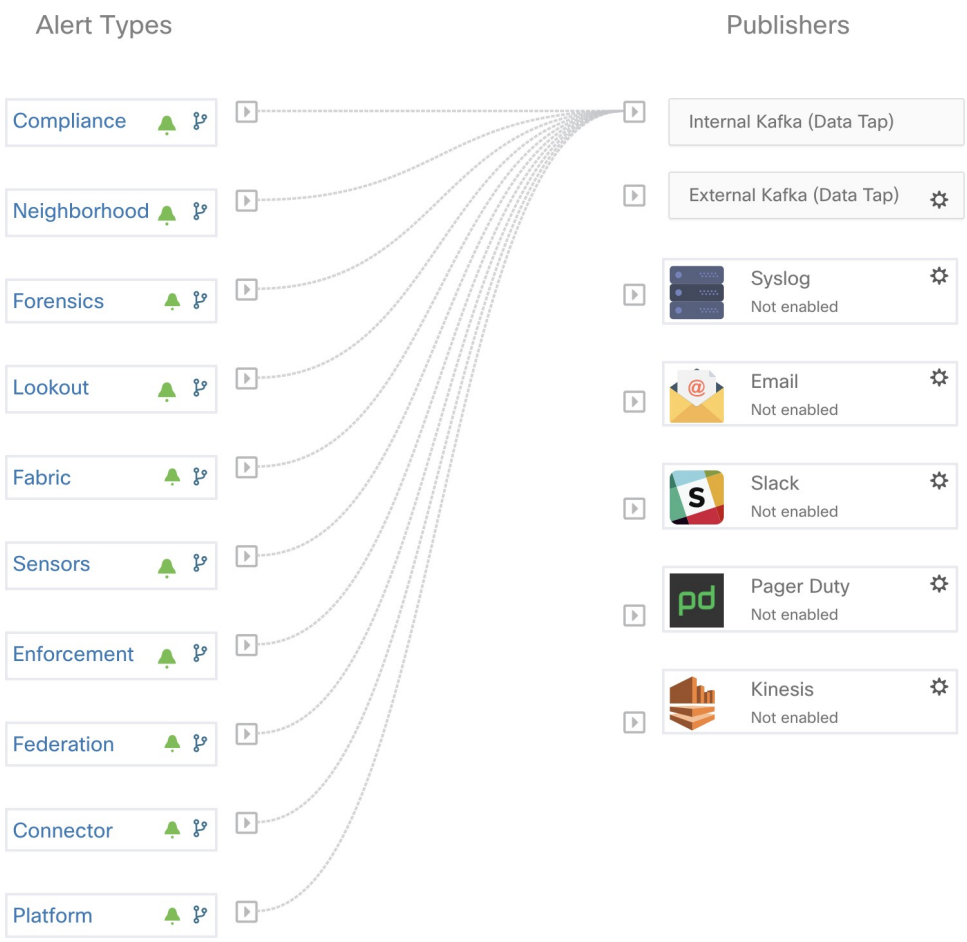


警告 個々のアラートを強制終了しないでください。基礎となるサービスがまだダウンしているか、稼働時間が予想されるしきい値を下回っているときに強制終了すると、次のアドミラル処理の反復で同じサービスに対して別のアラートが発生します。

アドミラル通知

アドミラルアラートのタイプは PLATFORM です。したがって、これらのアラートは、設定ページでの設定によるプラットフォームアラートへの適切な接続によって、さまざまなパブリッシャに送信されるように設定できます。利便性を考慮し、プラットフォームアラートと内部 Kafka 間の接続はデフォルトでオンになっています。これにより、アドミラルアラートが [現在のアラート（Current Alerts）] ページ（**[調査（Investigate）]** > **[アラート（Alerts）]** に移動）に表示されます。手動で設定する必要はありません。

図 *441* : プラットフォームアラートの設定



アドミラルアラートは、**[プラットフォーム（Platform）]** > **[クラスタの設定（Cluster Configuration）]** > **[アドミラルアラートメール（Admiral Alert Email）]** で設定された電子メールアドレスにも送信されます。

図 *442* : アドミラルメールのサンプル

There is a new admiral platform alert on your tetration cluster.
Service: Rpminstall
Start Time: 2020-07-14 23:09 UTC
Alert ID: 3
Description: Rpminstall uploads rpms into the cluster. Please look at /local/logs/tetration/rpminstall/rpm_upgrade.log for more details

This is an auto generated message about platform alerts on your cluster.
For more details, please go to Alerts On Cluster
Please make sure that you are on **Default Scope** to view the alerts.

そのため、ユーザーは TAN エッジアプライアンスをセットアップしていなくても、アドミラル通知を受け取ることができます。この動作は、以前のリリースの Bosun の動作に似ています。

図 *443 :* アドミラルメール



これらの電子メール通知は、[現在のアラート（Current Alerts）] ページと同じトリガーに基づいて生成されます。したがって、電子メール通知はアラートの作成時に送信され、UTC の午前 0 時に日次概要メールが送信されます。日次概要メールには、すべてのアクティブなアラートと過去 24 時間以内に閉じられたアラートが一覧表示されます。

図 *444 :* 概要アドミラルメールのサンプル

Daily summary of admiral platform alerts:

State:Active

Service: DataBackup
Start Time: 2020-07-14 21:58 UTC
Alert ID: 1
Description: The last successful checkpoint was over 48 hours ago.

State:Closed

Service: Rpminstall
Start Time: 2020-07-14 22:41 UTC
Alert ID: 2
Description: Rpminstall uploads rpms into the cluster. Please look at /local/logs/tetration/rpminstall/rpm_upgrade.log for more details

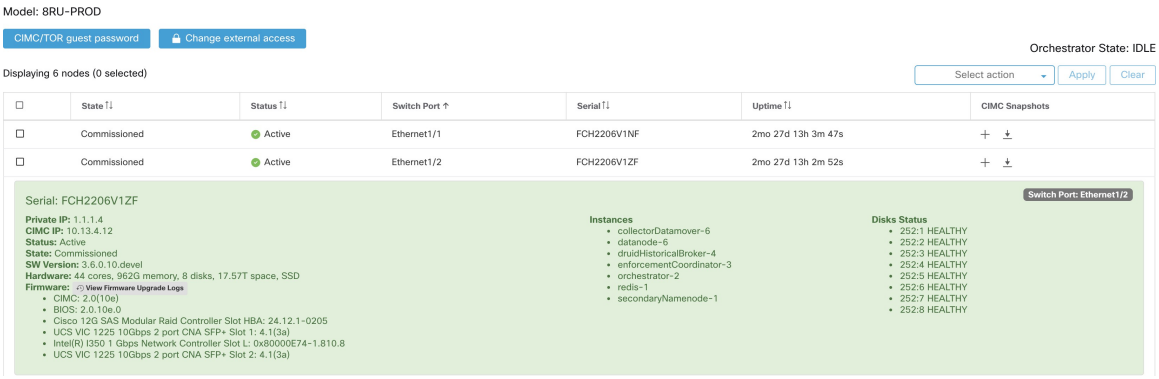
This is an auto generated message about platform alerts on your cluster.
For more details, please go to Alerts On Cluster
Please make sure that you are on **Default Scope** to view the alerts.

アクティブなアラートがなく、過去 24 時間以内に閉じられたアラートもない場合、電子メールノイズを減らすために概要メールはスキップされます。

クラスタのステータス

左側のナビゲーションバーの [トラブルシュート（Troubleshoot）] メニューにある [クラスタのステータス（Cluster Status）] ページには、**サイト管理者**ユーザーがアクセスできますが、アクションを実行できるのは**カスタマーサポート**ユーザーのみです。Cisco Secure Workload ラック内にあるすべての物理サーバーのステータスが表示されます。テーブルの各行は、ハードウェアとファームウェアの構成、CIMC IP アドレス（割り当てられている場合）などの詳細が設定された物理ノードを表します。行をクリックすると、ノードの詳細ビューを表示できます。このページでは、ノードの CIMC パスワードを変更し、ノードへの外部アクセスを有効/無効にすることもできます。 [クラスタのステータス（Cluster Status）] ページにはオーケストレータの状態も表示され、カスタマーサポートにコンテキストを提供できます。

図 *445* : クラスタのステータス



すべてのノードに影響するアクション

CIMCパスワードの変更と外部CIMCアクセスの有効化/無効化は、[CIMC/TORゲストパスワード（CIMC/TOR guest password）] ボタンおよび [外部アクセスの変更（Change external access）] ボタンを使用して行うことができます。これらのアクションはクラスタ内のすべてのノードに影響します。

外部 CIMC アクセスの詳細

[外部アクセスの変更（Change external access）] ボタンをクリックするとポップアップが開き、外部 CIMC アクセスのステータスが表示され、CIMC への外部アクセスを有効化、更新、または無効化できます。

[有効化（Enable）] ボタンをクリックすると、クラスタがバックグラウンドで構成され、外部 CIMC アクセスが有効になります。これらのタスクが完了し、外部 CIMC アクセスが完全に有効になるまでに最大 60 秒かかる場合があります。外部 CIMC アクセスが有効になっており、アクセスの自動期限切が設定されている場合、ポップアップが表示され、[有効（Enable）] ボタンが [更新（Renew）] に変わり、外部 CIMC アクセスを更新できることが反映されます。外部 CIMC アクセスを更新すると、有効期限が現在の時刻から 2 時間先になります。

外部CIMCアクセスが有効になっている場合、ノードの詳細（ノードの行をクリックして表示可能）のCIMC IPアドレスは、CIMC WebUI に直接アクセスできるクリック可能なリンクになります。このリンクを表示するには、[クラスタのステータス（Cluster Status）] ページのリロードが必要になる場合あります。

図 *446* : 外部 *CIMC* アクセスノードの詳細



CIMC WebUI には通常、自己署名証明書があり、CIMC WebUI にアクセスすると、証明書が無効であることを示すエラーがブラウザに表示される可能性があります。Google Chrome を使用

している場合、証明書チェックをバイパスしてCIMC WebUIにアクセスするためには、無効な証明書エラーが Google Chrome に表示されたときに、引用符なしで「thisisunsafe」と入力する必要があります。

CIMC WebUI では、CIMC バージョンが 4.1(1g) 以降の場合にのみ、KVM アクセスが機能します。外部 CIMC アクセスが有効になると、アクセスを更新または無効にしない限り、2 時間後に自動的に無効になります。

外部CIMCアクセスを無効にすると、クラスタがバックグラウンドで構成され、外部CIMCアクセスが無効になります。これらのタスクが完了し、外部CIMCアクセスが完全に無効になるまでに最大 60 秒かかる場合があります。

表 *34* : 物理ノードの詳細

フィールド	説明
[Status（ステータス）]	[ステータス（Status）] フィールドは、ノードの電源ステータスを示します。値は以下のとおりです。 - [アクティブ（Active）]：ノードの電源がオンになっています。 [非アクティブ（Inactive）]：ノードの電源が入っていないか、接続されていません。

フィールド	説明
[状態（State）]	[状態（State）] フィールドは、ノードのクラスタメンバーシップの状態を示します。値は以下のとおりです。 [新規（New）] : ノードはまだクラスタの一部ではありません。 [初期化済み（Initialized）] : ノードはクラスタの一部です。ただし、Cisco Secure Workload ソフトウェアはまだ完全にはノードにインストールされていません。 [稼働済み（Commissioned）] : ノードは Cisco Secure Workload ソフトウェアを使用して稼働しています。 [SW バージョン（SW Version）] フィールドも表示され、個々のノードのバージョンがクラスタ全体と同じでない場合は赤に変わります。 [稼働停止（Decommissioned）] : ノードはクラスタから削除されています（RMA の目的で）。ノードを新しいハードウェアと交換する必要があります。ノードは、デコミッションアクションにより稼働を停止できます。下記のアクションを参照してください。
[スイッチポート（Switch Port）]	物理ノードが接続されている 2 つのスイッチのスイッチポートを指します。
[稼働時間（Uptime）]	ノードが再起動またはシャットダウンせずに稼働していた時間を示します。
[CIMCスナップショット（CIMC Snapshots）]	CIMC テクニカルサポートデータの収集を開始して、ダウンロードするために使用できます。

表 *35* : アクション

アクション	説明
[コミッション（Commission）]	このアクションを選択すると、新しいノードがクラスタに組み込まれます。このアクションについては、状態が [新規（New）] のノードのみを選択できます。

アクション	説明
[デコミッション（Decommission）]	現在クラスタに属しているノードを削除するには、このアクションを選択します。このアクションについては、状態が [稼働済み（Commissioned）] または [初期化済み（Initialized）] のノードのみを選択できます。
[再イメージ化（Reimage）]	このアクションを選択すると、ボックス内に Secure Workload ソフトウェアが再インストールされます。これにより、ボックス内のファイルがすべて消去されます。ベアメタル オペレーティング システムを旧バージョンから新バージョンにアップグレードする際に特に便利です。この手順は、ベアメタルが稼働停止になった後に必要になります。
[ファームウェアのアップグレード（Firmware upgrade）]	ファームウェア情報は、CIMC IP に到達可能なノードで利用できます。このアクションは、旧バージョンのノードのファームウェアをアップグレードするのに役立ちます。
[電源オフ（Power off）]	ノードの電源を切るには、このアクションを選択します。ステータスが [非アクティブ（Inactive）] で**シャットダウン中**のノードの電源を切ることはできないので注意してください。

ファームウェア アップグレードの詳細

Secure Workload 物理アプライアンスには、ユニファイドコンピューティング システム（UCS）Cisco Integrated Management Controller（CIMC）ホストアップグレードユーティリティ（HUU）の ISO イメージがバンドルされています。[クラスタのステータス（Cluster Status）] ページでファームウェア アップグレード オプションを使用して、物理ベアメタルを Secure Workload RPM ファイルにバンドルされている HUU ISO に含まれる UCS ファームウェアのバージョンに更新できます。

ベアメタルホストは、ステータスが [アクティブ（Active）] または [非アクティブ（Inactive）] で、ベアメタルのステータスが [初期化（Initialized）] または [SKU不一致（SKU Mismatch）] でない場合に、ファームウェアの更新を開始できます。UCS ファームウェアを一度に更新できるベアメタルは 1 つだけです。ファームウェアの更新を開始するには、Secure Workload オーケストレータの状態が [アイドル（Idle）] である必要があります。UCS ファームウェアの更新が開始されると、Consul リーダー、アクティブなオーケストレータ、またはアクティブなファームウェアマネージャ（fwmgr）を他のホストに切り替える必要がある場合、[クラスタのステータス（Cluster Status）] ページに固有の UI 機能の一部が一時的に影響を受けることがあります。これらのスイッチオーバーは自動的に行われます。ファームウェアの更新中は、更新中のベアメタルホストのファームウェアの詳細は表示されません。更新が完了した後、[クラスタのステータス（Cluster Status）] ページにファームウェアの詳細が再度表示されるまで最大 15 分か

かることがあります。ファームウェアの更新を開始する前に、[サービスのステータス（Service Status）] ページですべてのサービスが正常であることを確認してください。

ベアメタルでファームウェアの更新を開始すると、fwmgr では更新が続行できることを確認し、必要に応じてベアメタルを正常にパワーダウンし、ベアメタルの CIMC にログインして HUU ベースのファームウェアの更新を開始します。この HUU ベースのファームウェアの更新プロセスには、HUU ISO でベアメタルを起動させ、更新を実行し、CIMC を再起動して新しいファームウェアをアクティブ化し、その後 HUU ISO でベアメタルを再起動して、更新が完了したことを確認することが含まれます。全体的な更新プロセスには、G1 ベアメタルの場合は 2 時間以上、G2 ベアメタルの場合は 1 時間以上かかる場合があります。ファームウェアの更新プロセスが開始されると、ベアメタルと、そのベアメタルで実行されているすべての仮想マシンがクラスタ内でアクティブでなくなるため、[サービスのステータス（Service Status）] ページに、一部のサービスが正常でないと示される場合があります。ファームウェアの更新が完了すると、ベアメタルがクラスタ内で再びアクティブになるまでにさらに 30 分かかり、すべてのサービスが再び正常になるまでにさらに時間がかかる場合があります。ファームウェアの更新後 2 時間以内にサービスが回復しない場合は、シスコテクニカルサポートにお問い合わせください。

[クラスタのステータス（Cluster Status）] ページで、ベアメタルノードをクリックして、ベアメタルに関する詳細を展開できます。ファームウェアの更新が開始されたら、[ファームウェアのアップグレードログを表示（View Firmware Upgrade Logs）] ボタンをクリックして、ファームウェア更新のステータスを表示できます。このログには、ファームウェア更新の全体的なステータスが一番上に表示されます。内容は次のいずれかです。

- [ファームウェアの更新がトリガーされました（Firmware update has been triggered）]：ファームウェアの更新が要求されましたが、まだ開始されていません。このステータス中に、fwmgr では、ファームウェアの更新に必要なサービスが機能していること、および CIMC がそれらのサービスに到達できることが確認されます。
- [ファームウェアの更新を実行中です（Firmware update is running）]：ファームウェアの更新が開始されました。ファームウェアの更新がこの状態に達すると、CIMC と HUU で更新が制御され、Secure Workload クラスタでは CIMC から取得した更新に関するステータスが報告されます。
- [ファームウェアの更新がタイムアウトしました（Firmware update has timed out）]：ファームウェアの更新の一部のプロセスが、完了予測時間を超えたことを示します。[ファームウェアの更新を実行中です（Firmware update is running）] のフェーズに入ると、ファームウェアの更新プロセス全体の制限時間は 240 分になります。ファームウェアの更新中に、新しいバージョンでリブートすると CIMC が到達不能になることがあります。この到達不能状態のタイムアウトは、ファームウェアの更新が「タイムアウト」と宣言されるまでの 40 分間です。ファームウェアの更新が開始されると、その更新のモニタリングは 120 分後にタイムアウトします。
- [ファームウェアの更新がエラーのため失敗しました（Firmware update has failed with an error）]：エラーが発生し、ファームウェアの更新が失敗したことを示します。通常、CIMC では成功または失敗は示されません。そのため、この状態は通常、ファームウェアの更新が実際に実行される前にエラーが発生したことを示しています。

- [ファームウェアの更新が終了しました（Firmware update has finished）]：ファームウェアの更新は、エラーやタイムアウトが発生することなく終了しました。通常、CIMC では成功または失敗は示されないため、[クラスタのステータス（Cluster Status）] ページでこれらの詳細が確認できるようになった後に、UCS ファームウェアバージョンが更新されているか確認することをお勧めします。詳細が確認できるようになるまで最大 15 分かかります。

[ファームウェアのアップグレードログを表示（View Firmware Upgrade Logs）] ポップアップウィンドウの全体的なステータスの下にある [更新の進行状況（Update progress）] セクションには、ファームウェア更新の進行状況を示すタイムスタンプ付きのログメッセージが含まれます。これらのログメッセージに [ホストの再起動が進行中です（Rebooting Host In Progress）] ステータスが表示されると、CIMC で更新が制御され、クラスタがその更新をモニターします。後続のほとんどのログメッセージはCIMCから直接送信され、更新のステータスが変更された場合にのみログメッセージのリストに追加されます。

CIMC で個々のコンポーネント更新ステータスの提供が開始されると、[ファームウェアのアップグレードログを表示（View Firmware Upgrade Logs）] ポップアップの [更新の進行状況（Update progress）] セクションの下に、[コンポーネントの更新ステータス（Component update status）] セクションが表示されます。このセクションでは、ベアメタル上のさまざまな UCS コンポーネントの更新のステータスがすぐに把握できます。

データのバックアップと復元

データのバックアップと復元は、Cisco Secure Workload クラスタ、コネクタ、および外部オーケストレータからオフサイトストレージにデータをコピーするディザスタ リカバリ メカニズムです。災害が発生した場合は、このオフサイトストレージから同じフォームファクタのクラスタにデータが復元されます。データのバックアップと復元のオプションを使用すると、災害が発生した場合に別のバックアップサイトに切り替えることができます。

- データのバックアップと復元は、物理クラスタ（8RU と 39RU の両方）でサポートされています。
- データは、S3V4 API と互換性のある任意の外部オブジェクトストアにバックアップできます。
- Cisco Secure Workload には、データをバックアップするための十分な帯域幅とストレージが必要です。
- 継続的モードを使用した完全バックアップには、200 TB のストレージが推奨されます。必要な実際のストレージ容量を判断するには、[データバックアップ（Data Backup）] ページで使用可能な [キャパシティプランナー（Capacity Planner）] オプションを使用できます。詳細については、キャパシティプランナーの使用（837ページ）を参照してください。複数のバックアップ用のストレージが不足していると、ストレージの制限内で機能するように、古いバックアップが頻繁に削除されます。少なくとも 1 つのバックアップのために十分なストレージが必要です。

- データは、プライマリと同じバージョンを実行している互換性のあるフォームファクタのクラスタにのみ復元できます。たとえば、8RU クラスタからのデータは、別の 8RU にのみ復元できます。

データ バックアップ

データバックアップのスケジュールは、UI の [データバックアップ（Data Backup）] セクションを使用して設定できます。バックアップは、設定に基づいて 1 日に 1 回スケジュールされた時刻にトリガーされるか、継続的に実行されるように設定することができます。バックアップの成功は、チェックポイントと呼ばれます。チェックポイントは、クラスタのプライマリデータストアのポイント イン タイム スナップショットです。

成功したチェックポイントを使用して、データを別のクラスタまたは同じクラスタに復元できます。

クラスタ設定データは、すべてのチェックポイントで常にバックアップされます。フローおよびその他のデータが、バックアップされるデータの大部分を占めます。そのため、適切に設定されている場合は、増分変更のみがバックアップされます。増分バックアップは、外部ストレージにプッシュされるデータの量を減らすために役立ち、ネットワークの過負荷を回避できます。増分バックアップが設定されている場合、必要に応じて、すべてのデータソースに対してスケジュールに従って完全バックアップをトリガーすることができます。完全バックアップでは、チェックポイント内のすべてのオブジェクトがコピーされます。オブジェクトが変更されていない場合でもコピーされます。これにより、クラスタ、クラスタとオブジェクトストア間のネットワーク、およびオブジェクトストア自体にかなりの負荷がかかる可能性があります。オブジェクトが破損している場合、またはオブジェクトストアに回復不能なハードウェア障害がある場合は、完全バックアップが必要になることがあります。さらに、バックアップ用に提供されているバケットが変更された場合は、完全バックアップが自動的に適用されます。これは、増分バックアップを役立てるには事前に完全バックアップが必要なためです。

表 *36 :* 異なるモードでのバックアップされるクラスタデータ

Cisco Secure Workload クラスタデータ	**完全バックアップモードでデータがバックアップされるか**	**リーンモードでデータがバックアップされるか**
クラスタ設定	**対応**	**対応**
クラスタのイメージングに使用される RPM	**対応**	**対応**
ソフトウェアエージェント展開イメージ	**対応**	**対応**
フロー データベース	**対応**	×
自動ポリシー検出に必要なデータ	**対応**	×

Cisco Secure Workload クラスタデータ	完全バックアップモードでデータがバックアップされるか	リーンモードでデータがバックアップされるか
適用ポリシー	対応	×
ファイルハッシュ、データリークモデルなどのフォレンジックに役立つデータ	対応	×
攻撃対象領域の分析に役立つデータ	対応	×
CVE データベース	対応	×

（注） セキュアコネクタ情報は、オンプレミスバージョンの Cisco Secure Workload ではバックアップまたは復元されませんが、SaaS バージョンの Cisco Secure Workload ではバックアップおよび復元されます。

データバックアップの前提条件

- Data Backup and Restore（DBR）機能のアクティベーションキーを取得するには、taentitlement@cisco.com に電子メールを送信して DBR アクティベーションキーを要求します。電子メールにはクラスタ ID ファイルも添付します。

（注） ライセンスの権限付与は、プライマリ（アクティブ）クラスタにのみ必要です。スタンバイクラスタには必要ありません。

- オブジェクトストアのアクセスキーと秘密鍵が必要です。データのバックアップと復元のオプションは、オブジェクトストアの事前認証されたリンクでは機能しません。
- Cisco Secure Workload アプライアンスがオブジェクトストアに使用する帯域幅を調整するポリシングを設定します。バックアップするデータの量が多い場合に低帯域幅でポリシングすると、バックアップが失敗する可能性があることに注意してください。
- クラスタの FQDN を設定し、ソフトウェアエージェントが FQDN を解決できることを確認します。

（注） データのバックアップと復元を有効にすると、現在および以降のソフトウェア エージェント バージョンのみをインストールおよびアップグレードに使用できます。現在のクラスタバージョンよりも前のバージョンは、互換性がないため非表示のままです。

ソフトウェアエージェントまたは Kafka の FQDN の要件

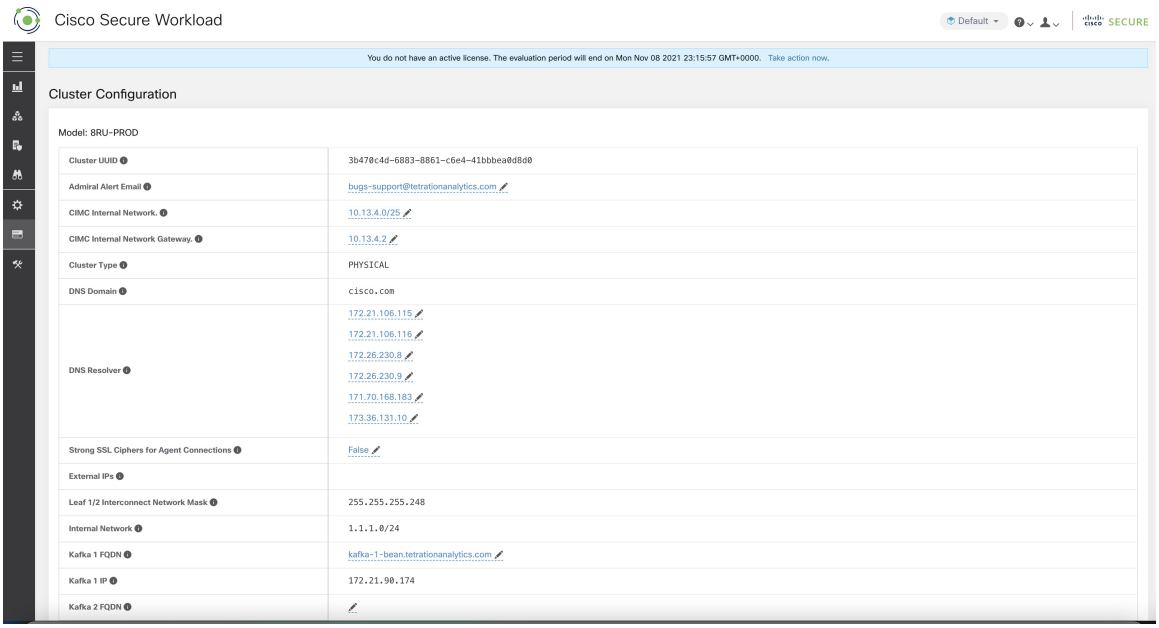
ソフトウェアエージェントは、IP アドレスを使用して Cisco Secure Workload アプライアンスから制御情報を取得します。データのバックアップと復元を有効にして、災害後のシームレスなフェールオーバーを可能にするため、エージェントを FQDN の使用に切り替える必要があります。このスイッチでは、Cisco Secure Workload クラスタのアップグレードだけでは不十分です。ソフトウェアエージェントは、Cisco Secure Workload バージョン 3.3 以降で FQDN の使用をサポートしています。そのため、エージェントのフェールオーバーを有効にして、データのバックアップと復元の準備ができていることを確認するには、エージェントをバージョン 3.3 以降にアップグレードします。

FQDN が設定されていない場合、デフォルトの FQDN は次のとおりです。

IP タイプ（IP Type）	デフォルトの FQDN
センサー VIP	wss{{cluster_ui_fqdn}}
Kafka 1	kafka-1-{{cluster_ui_fqdn}}
Kafka 2	kafka-2-{{cluster_ui_fqdn}}
Kafka 3	kafka-3-{{cluster_ui_fqdn}}

FQDN は、**[プラットフォーム（Platform）]** > **[クラスタ設定（Cluster Configuration）]** ページで変更できます。

図 *447 : [クラスタ設定（Cluster Configuration）]* ページでのデータのバックアップと復元の *FQDN* または *IP*



FQDN の DNS レコードを、同じページで提供される IP で更新します。次の表に、IP と FQDN のマッピングを示します。

フィールド名	対応する IP フィールド	説明
センサー VIP FQDN	センサー VIP	FQDN を更新してクラスタ コントロール プレーンに接続する
Kafka 1 FQDN	Kafka 1 IP	Kafka ノード 1 IP
Kafka 2 FQDN	Kafka 2 IP	Kafka ノード 2 IP
Kafka 3 FQDN	Kafka 3 IP	Kafka ノード 3 IP

（注） センサー VIP および Kafka ホストの FQDN は、データのバックアップと復元が設定される前にのみ変更できます。設定後は、FQDN は変更できません。

オブジェクトストアの要件

オブジェクトストアは、S3V4 準拠のインターフェイスを提供する必要があります。

- **所在地（Location）**

  オブジェクトストアの場所は、ストアからのバックアップと復元に伴う遅延にとって重要です。復元時間を短縮するには、オブジェクトストアがスタンバイクラスタの近くにあることを確認します。

- **Bucket**

  オブジェクトストアに、Cisco Secure Workload の専用の新しいバケットを作成します。クラスタのみが、このバケットへの書き込みアクセス権を持つ必要があります。このバケットに対して、クラスタがオブジェクトを書き込み、保持を管理します。バケット用に少なくとも 200 TB のストレージをプロビジョニングし、バケットのアクセスと秘密鍵を取得します。Cisco Secure Workload でのデータのバックアップと復元は、事前認証されたリンクでは機能しません。

（注） オブジェクトストアとして Cohesity が使用されている場合は、スケジュール時にマルチパートアップロードを無効にします。

- **HTTPS**

  データ バックアップ オプションは、オブジェクトストアでの HTTPS インターフェイスのみをサポートします。オブジェクトストアへ転送中のデータが暗号化され、安全であることを保証するためです。ストレージ SSL/TSL 証明書が信頼できるサードパーティ CA によって署名されている場合、クラスタはその証明書を使用してオブジェクトストアを認証します。オブジェクトストアが自己署名証明書を使用している場合は、[サーバーCA証明

書を使用（Use Server CA Certificate）] オプションを選択して、公開キーまたは CA をアップロードできます。

- **サーバー側の暗号化**

  Cisco Secure Workload クラスタに割り当てられたバケットのサーバー側の暗号化をオンにすることを強く推奨します。クラスタは、HTTPS を使用してオブジェクトストアにデータを転送します。ただし、オブジェクトストアはオブジェクトを暗号化して、保存されたデータの安全性を確保する必要があります。

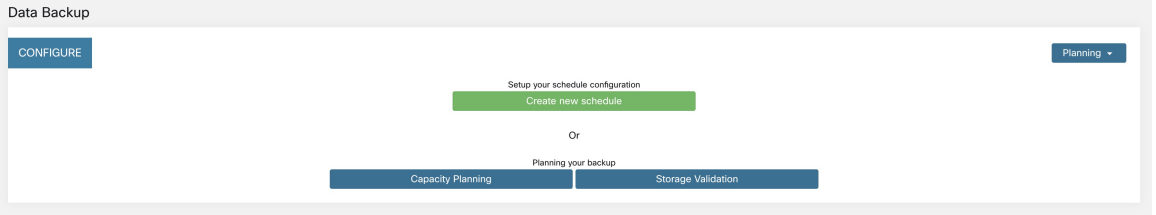
データのバックアップの設定

Cisco Secure Workload でデータバックアップを設定するには、次の手順を実行します。

1. **計画**：データバックアップのオプションでは、プランナーを使用して、オブジェクトストアへのアクセスをテストし、ストレージ要件と、各日に必要なバックアップ期間を決定することができます。これは、スケジュールを設定する前の試験に使用できます。

   データのバックアップと復元の計算ツールを使用するには、**[プラットフォーム（Platform）]>[データバックアップ（Data Backup）]** に移動します。データのバックアップと復元が設定されていない場合は、データバックアップのランディングページに移動します。

   図 *448*：バック アップランディング ページ

   - ストレージプランナーの使用 （836 ページ）
   - キャパシティプランナーの使用 （837 ページ）

（注） [プラットフォーム（Platform）] の下に [データバックアップ（Data Backup）] オプションが表示されない場合は、データのバックアップと復元を有効にするライセンスがあることを確認してください。

2. **データバックアップの設定とスケジューリング**：Cisco Secure Workload は、設定された時間枠でのみデータをオブジェクトストアにコピーします。バックアップを初めて設定するときに、事前チェックが実行され、FQDN が解決可能であり、正しい IP に解決されることを確認します。最初の検証後に、登録済みのソフトウェアエージェントに更新がプッシュされ、FQDN の使用に切り替わります。FQDN がないと、エージェントはディザスタイベント後に別のクラスタにフェールオーバーできません。FQDN の使用をサポートするに

は、クラスタでサポートされている最新バージョンにエージェントをアップグレードする必要があり、すべてのエージェントがセンサー VIP FQDN を解決できる必要があります。Cisco Secure Workload リリース 3.3 以降では、優れた可視性エージェントと適用エージェントのみがデータのバックアップと復元をサポートしており、FQDN の使用に切り替えます。

スケジュールを作成し、データバックアップを設定するには、データバックアップの設定（838 ページ）を参照してください。

ストレージプランナーの使用

ストレージプランナーを使用するには、次の手順を実行します。

手順

ステップ 1 ストレージに Cisco Secure Workload との互換性があることを確認するには、次のいずれかのアクションを実行します。

- [データバックアップ（Data Backup）] ランディングページで、[ストレージプランニング（Storage Planning）] をクリックします。
- [プランニング（Planning）] ドロップダウンメニューから、[ストレージ（Storage）] を選択します。

[ストレージプランニング（Storage Planning）] ページが表示されます。

ステップ 2 次の詳細を入力します。

- ストレージの名前。
- S3 準拠のストレージエンドポイントの URL。
- ストレージで設定されている S3 準拠のバケット名。
- （特定のストレージのオプション）S3 準拠のストレージのリージョン。
- ストレージへのアクセスキー。
- ストレージの秘密鍵。

ステップ 3 （オプション）必要に応じて、HTTP プロキシを有効にできます。

ステップ 4 （オプション）バックアップされたデータのマルチパートアップロードを使用するには、[マルチパートアップロードの使用（Use Multipart Upload）] を有効にします。

ステップ 5 （オプション）ストレージサーバーの認証に CA 証明書が必要な場合は、[サーバーCA証明書の使用（Use Server CA Certificate）] を有効にして、証明書の詳細を入力します。

ステップ 6 [テスト（Test）] をクリックします。

ストレージの検証では、次のことがテストされます。

- オブジェクトストアとバケットに対して認証およびアクセスします。
- 設定されたバケットにアップロードし、そのバケットからダウンロードします。
- 帯域幅をチェックします。

ストレージ プランニング プロセスが完了するまでに、約 5 分かかる可能性があります。

キャパシティプランナーの使用

キャパシティプランナーを使用するには、次の手順を実行します。

手順

ステップ 1 想定されるストレージサイズとバックアップ時間を計画するために、次のいずれかのアクションを実行します。

- [データバックアップ（Data Backup）] ランディングページで、[□キャパシティプランニング（Capacity Planning）] をクリックします。
- [プランニング（Planning）] ドロップダウンメニューから、[キャパシティ（Capacity）] を選択します。

[キャパシティプランニング（Capacity Planning）] ページが表示されます。

ステップ 2 データをバックアップするための最大帯域幅制限を入力します。

この帯域幅は、オブジェクトストアへのデータをスロットリングするポリサー設定の値以下である必要があります。

ステップ 3 登録済みソフトウェアエージェント数は自動的に入力されます。予測に基づいて、エージェント数を変更できます。

ステップ 4 （オプション）設定データ以外をバックアップから除外するには、[リーンデータモード（Lean Data Mode）] を有効にします。このオプションを使用すると、ストレージの制限が 75% 軽減されます。

ステップ 5 ストレージバケットに設定される最大ストレージ。これにより、バックアップの保持期間が自動的に設定されます。

必要な詳細を入力すると、[推定バックアップ期間（Estimated Backup Duration）] に 1 日のデータのバックアップに必要な時間が表示されます。この値は、一般的なエージェント負荷、推定エージェント数、および設定された最大帯域幅に基づく推定値です。[推定最大ストレージ（Estimated Maximum Storage）] には、指定された保持と推定エージェント数をサポートするために Cisco Secure Workload で必要となる最大ストレージの推定値が表示されます。

データバックアップの設定

手順

ステップ 1 データバックアップのランディングページで、[新しいスケジュールの作成（Create new schedule）] をクリックします。

ステップ 2 実行する前提条件チェックを確認するには、[承認（Approve）] ボタンをオンにして、[続行（Proceed）] をクリックします。

前提条件チェックは完了するまでに約 30 分かかり、スケジュールが初めて設定されたときにのみ実行されます。

図 *449* : バックアップの警告 - *FQDN* が設定されていることを確認します。

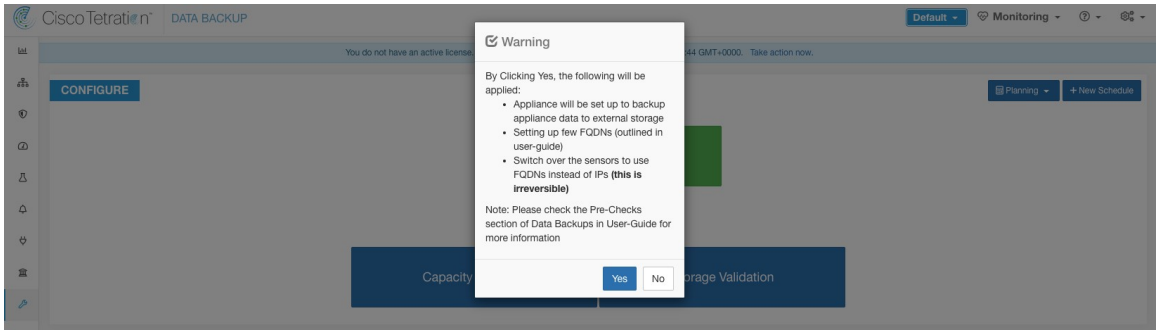
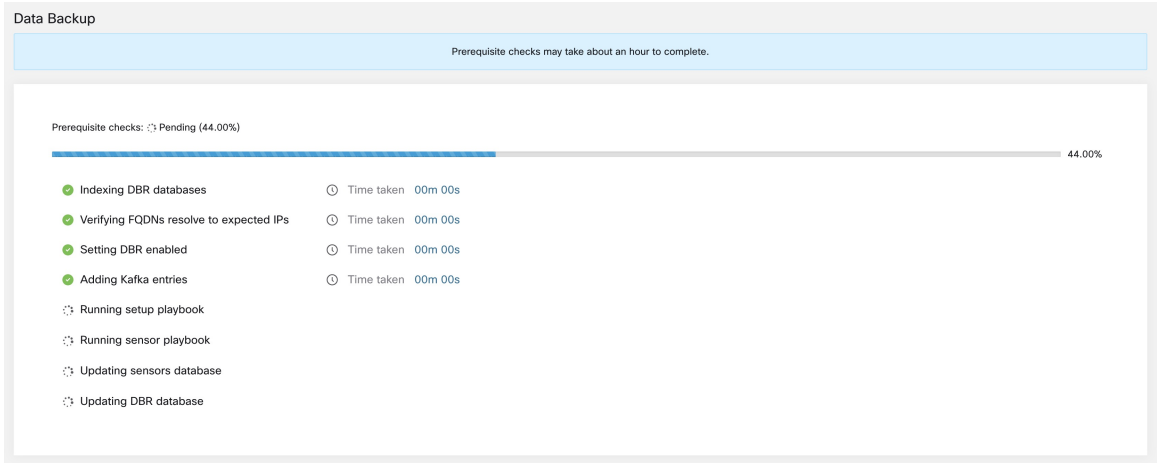


図 *450* : バックアップの前提条件の実行

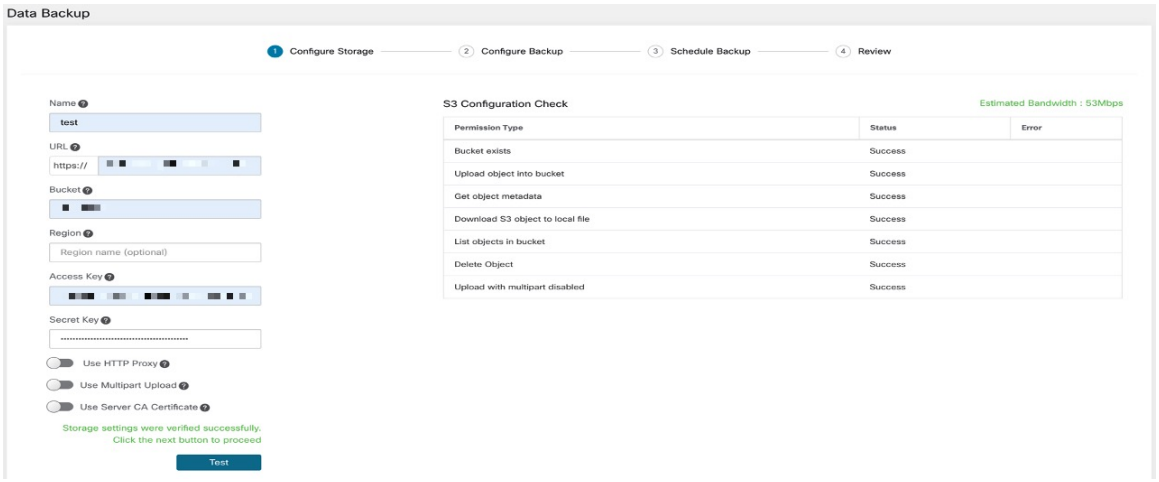


ステップ 3 ストレージを設定するには、次の詳細を入力して、[テスト（Test）] をクリックします。

- ストレージの名前。
- S3 準拠のストレージエンドポイントの URL。
- ストレージで設定されている S3 準拠のバケット名。

- （特定のストレージのオプション）S3 準拠のストレージのリージョン。
- ストレージへのアクセスキー。
- ストレージの秘密鍵。
- （オプション）必要に応じて、HTTP プロキシを有効にします。
- （オプション）バックアップされたデータのマルチパートアップロードを使用するには、[マルチパートアップロードの使用（Use Multipart Upload）] を有効にします。
- （オプション）ストレージサーバーの認証に CA 証明書が必要な場合は、[サーバーCA証明書の使用（Use Server CA Certificate）] を有効にして、証明書の詳細を入力します。

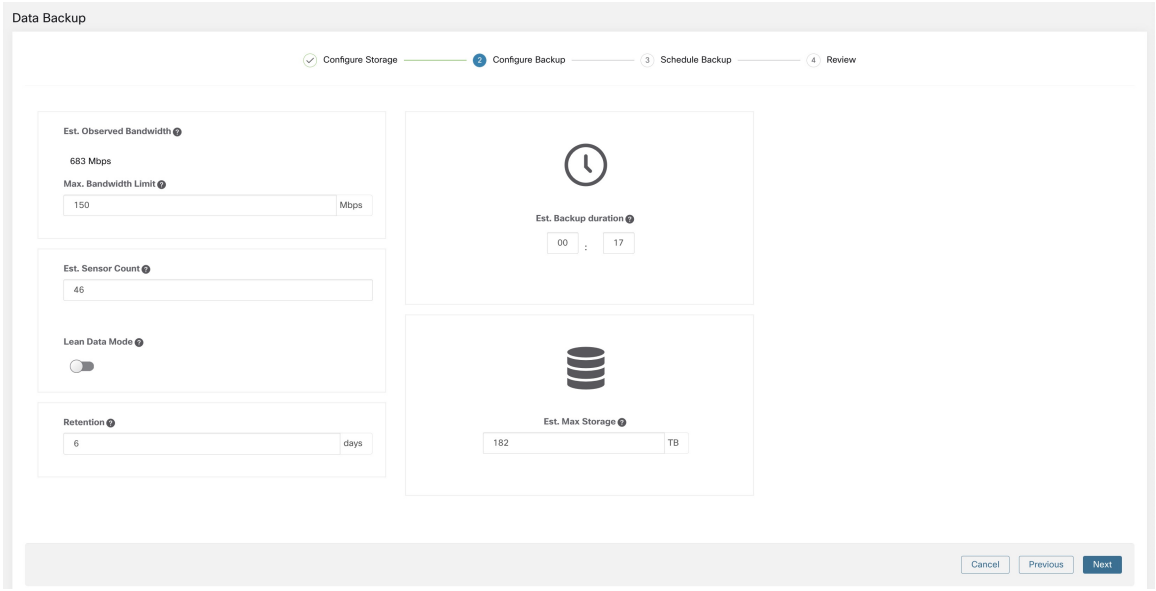
図 *451* : ストレージの設定



ステップ 4 ストレージ容量を設定するには、次の詳細を入力します。

- データをバックアップするための最大帯域幅制限。この帯域幅は、オブジェクトストアへのデータをスロットリングするポリサー設定の値以下である必要があります。
- 登録済みソフトウェアエージェント数は自動的に入力されます。予測に基づいて、エージェント数を変更できます。
- （オプション）設定データ以外をバックアップから除外するには、[リーンデータモード（Lean Data Mode）] を有効にします。このオプションを使用すると、ストレージの制限が 75% 軽減されます。
- ストレージバケットに設定される最大ストレージ。これにより、バックアップの保持期間が自動的に設定されます。

図 *452 :* キャパシティ プランニング



ステップ 5 バックアップをスケジュールするには、以下を有効にします。

- デフォルトでは、[バックアップの開始点を今日からに設定（Set starting backup point from today）] が有効になっています。このオプションでは、設定日の午前 0 時（UTC）より前に作成されたすべてのファイルが無視されます。稼働しているクラスタでは、初日にバックアップされるデータが大量に存在する場合があり、クラスタ、ネットワーク、およびオブジェクトストアに過剰な負荷がかかる可能性があります。既存のすべてのデータをバックアップする場合は、このチェックボックスを無効にします。ただし、ネットワーク、オブジェクトストア、およびクラスタへの影響に注意してください。

  （注） このオプションに関係なく、すべての設定データがバックアップされます。

- [継続的なバックアップ（Continuous backup）] : 有効にすると、前回のバックアップが完了してから 15 分後にデータがバックアップされます。このオプションを使用すると、バックアップを特定の時刻にスケジュールするのではなく、継続的に実行することができます。[継続的なバックアップ（Continuous backup）] が有効になっている場合、[タイムゾーン（Time zone）] および [許可されるバックアップ開始時間（Allowed Start backup window）] のオプションは使用できません。

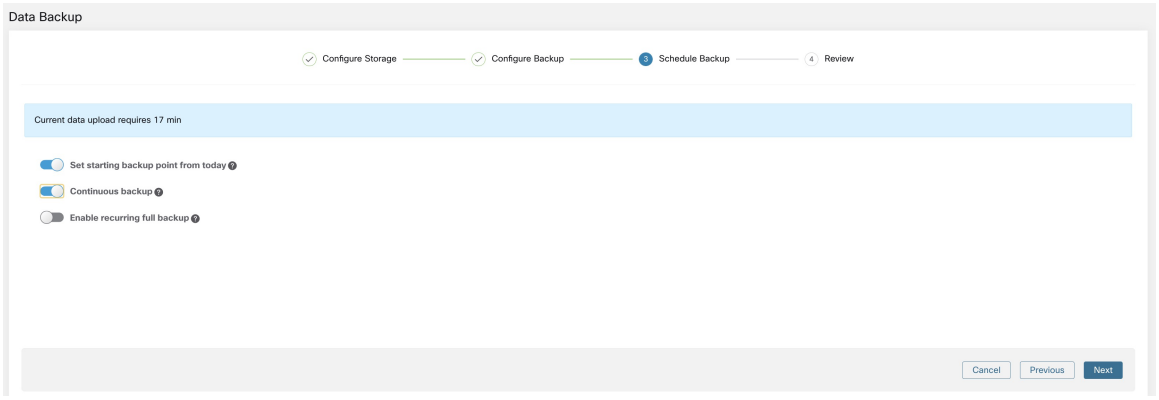
- 次の 2 つのオプションは、継続的なバックアップが使用されていない場合に、バックアップのスケジュールを設定するために使用されます。

  - [タイムゾーン（Time zone）] : デフォルトでは、Web ブラウザのタイムゾーンです。

  - [許可されるバックアップ開始時間（Allowed Start backup window）] : バックアップが開始される時刻（時間または分）。時刻は 24 時間形式で入力する必要があります。

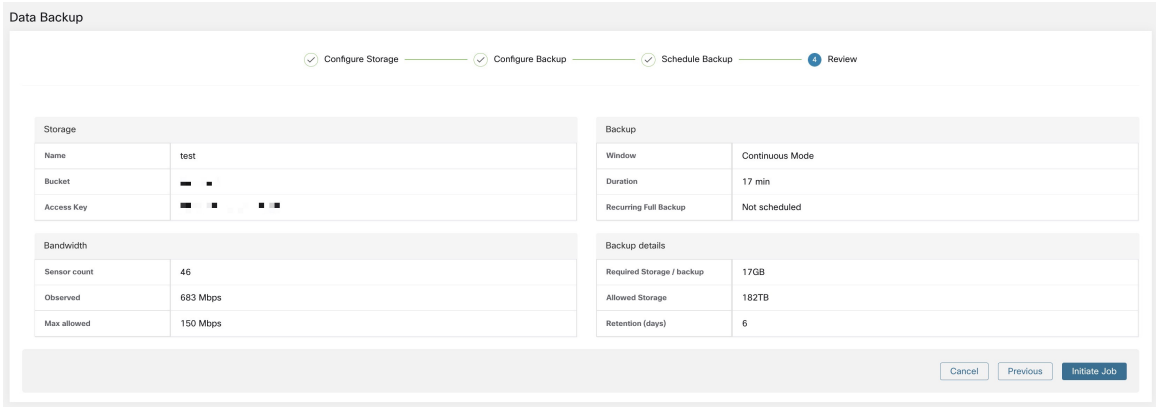
  - [定期的な完全バックアップの有効化（Enable recurring full backup）]（デフォルトでは無効）：有効にすると、完全バックアップのスケジュールを設定できます。デフォル

トでは、最初の完全バックアップの後は、すべてのバックアップが増分バックアップになります。この設定を有効にすると、指定されたスケジュールで完全バックアップが強制的に実行されます。



ステップ 6 設定済みのバックアップスケジュールと設定を確認し、[ジョブの開始（Initiate Job）] をクリックします。

図 *453* : バックアップ設定のレビュー



バックアップステータス

データバックアップの設定後は、継続的なモードが有効になっていない限り、スケジュールされた時刻に毎日バックアップがトリガーされます。バックアップのステータスは、**[プラットフォーム（Platform）] > [データバックアップ（Data Backup）]** に移動することで、[データバックアップ（Data Backup）] ダッシュボードで確認できます。

図 *454* : バックアップ/チェックポイントの状態

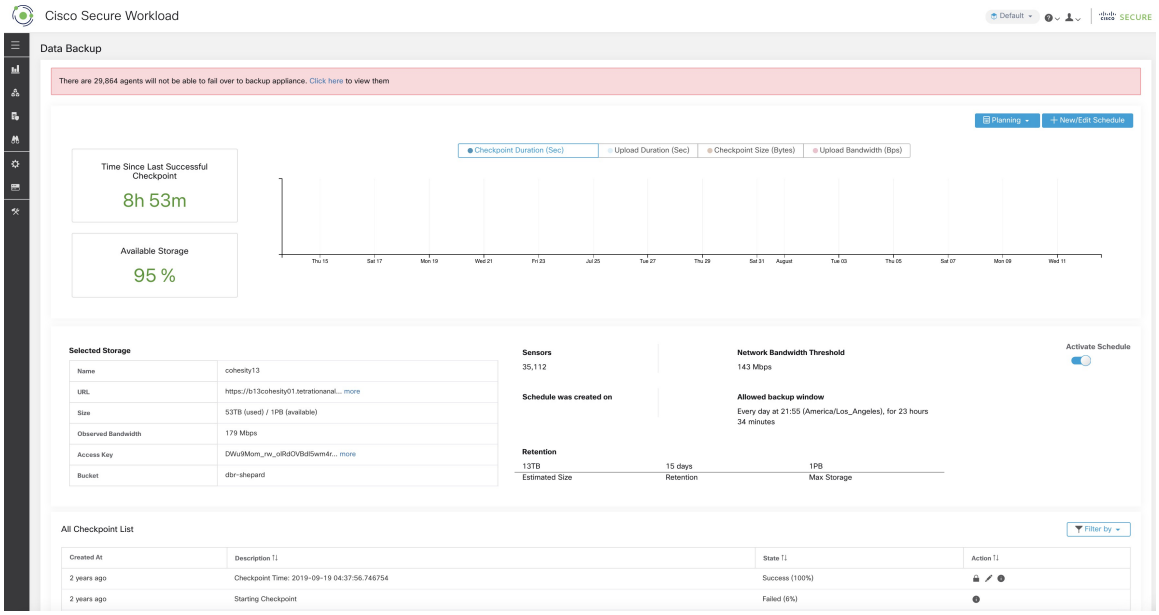
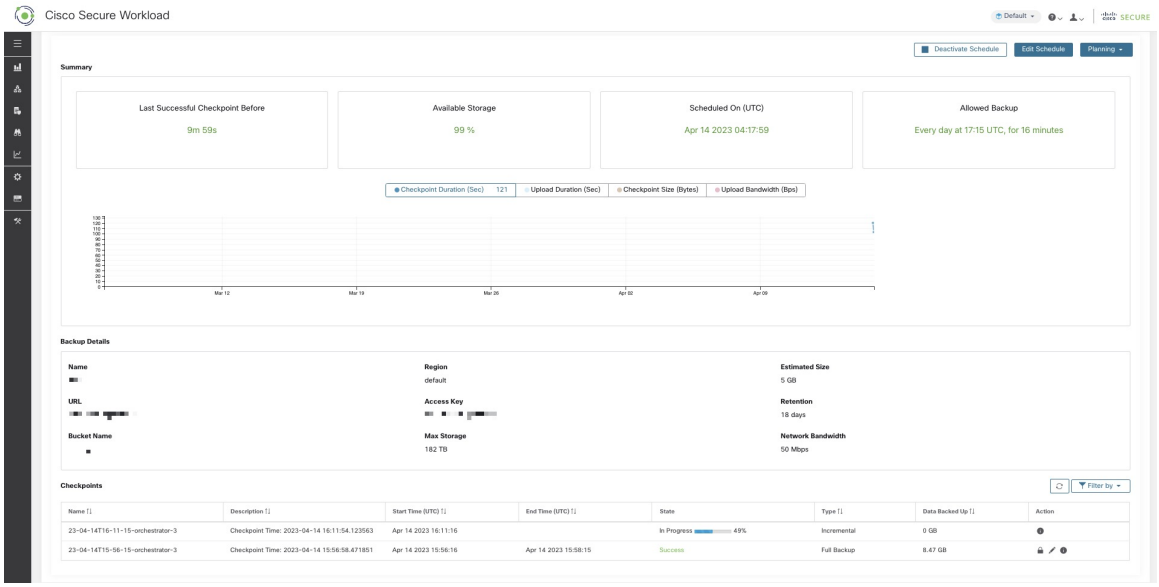


図 *455* : バックアップステータス



最後に成功したチェックポイントからの経過時間は、チェックポイントにかかる時間と 24 時間を足した時間未満である必要があります。たとえば、チェックポイントとバックアップに約 6 時間かかる場合、最後に成功したチェックポイントからの経過時間は 30 時間未満である必要があります。

次のグラフでは、追加情報を提供します。

- [チェックポイント期間（Checkpoint Duration）]：このグラフには、チェックポイントにかかる時間の近似曲線が表示されます。

- [アップロード時間（Upload Duration）]：このグラフには、チェックポイントをバックアップにアップロードするためにかかる時間の近似曲線が表示されます。
- [チェックポイントサイズ（Checkpoint Size）]：このグラフには、チェックポイントのサイズの近似曲線が表示されます。
- [アップロード帯域幅（Upload Bandwidth）]：このグラフには、アップロード帯域幅の近似曲線が表示されます。

この表は、すべてのチェックポイントを示しています。チェックポイントラベルは編集可能で、スタンバイクラスタでデータを復元するためにチェックポイントを選択するときにラベルを使用できます。

チェックポイントは、複数の状態に移行します。使用される状態を以下に示します。

- [作成済み/保留中（Created/Pending）]：チェックポイントは作成済みで、コピーされるのを待機しています。
- [実行中（Running）]：データが、外部ストレージにアクティブにバックアップされています。
- [成功（Success）]：チェックポイントが完了し、成功しました。データ復元に使用できます。
- [失敗（Failed）]：チェックポイントは完了しましたが、失敗しました。データ復元には使用できません。
- [削除中/削除済み（Deleting/Deleted）]：期限切れのチェックポイントが削除中であるか、削除されました。

スケジュールまたはバケットを変更するには、[スケジュールの編集（Edit Schedule）]をクリックします。ウィザードを完了するには、「データバックアップの設定」の項を参照してください。

チェックポイントの作成中のエラーをトラブルシューティングするには、トラブルシューティング：データのバックアップと復元 （849 ページ）を参照してください。

バックアップスケジュールの非アクティブ化

[スケジュールの非アクティブ化（Deactivate Schedule）]ボタンをクリックすることで、バックアップを非アクティブ化できます。スケジュールを変更する前に、バックアップスケジュールを非アクティブ化することをお勧めします。スケジュールを無効にするのは、進行中のチェックポイントがない場合のみにします。チェックポイントの進行中にテストを実行したり、スケジュールを無効にしたりすると、進行中のチェックポイントが失敗し、アップロードが未定義の状態になる可能性があります。

オブジェクトストアの保持

Cisco Secure Workload クラスタは、バケット内のオブジェクトのライフサイクルを管理します。バケットのオブジェクトを削除または追加してはなりません。これを行うと、不整合が発生

し、正常なチェックポイントが破損する可能性があります。構成ウィザードで、使用する最大ストレージを指定する必要があります。Cisco Secure Workload は、バケットの使用量が設定された制限内に収まるようにします。オブジェクトをエージアウトしてバケットから削除するストレージ保持サービスがあります。ストレージ使用量は、設定された最大ストレージと受信データレートに基づいて計算されます。保持サービスは、使用量がしきい値（バケット容量の80%）に達すると、使用量をしきい値未満に減らすために、保存されていないチェックポイントを削除しようとします。また、保持サービスでは、常に最低2つの成功したチェックポイントと、保存されたすべてのチェックポイントの、いずれか多い方が維持されます。保持サービスでチェックポイントを削除して容量を空けることができない場合、**チェックポイントでエラーが発生し始めます**。

チェックポイントの保持

新しいチェックポイントが作成されると、古いチェックポイントはエージアウトになり、削除されます。ただし、チェックポイントを保持することができ、保持設定により削除されることがなくなります。保持されたチェックポイントは削除されません。保持されたチェックポイントが複数ある場合、ある時点で新しいオブジェクト用のストレージが足りなくなりますが、エージアウトしたチェックポイントは保持されているため削除されません。ベストプラクティスとして、必要に応じてチェックポイントを保持し、参照用としてラベルにその理由と妥当性を含めてチェックポイントを更新します。チェックポイントを保持するには、必要なチェックポイントのロックアイコンをクリックします。

データの復元

データの復元オプションは、左側のナビゲーションバーの [プラットフォーム（Platform）] メニューで使用できます。

バックアップデータを使用してクラスタを復元するには、クラスタが **DBR スタンバイモード**になっている必要があります。現在、クラスタは**初期設定時にのみ**スタンバイモードに設定できます。

次の組み合わせが可能です。

プライマリクラスタ **SKU**	スタンバイクラスタ **SKU**
8RU-PROD	8RU-PROD、8RU-M5
8RU-M5	8RU-PROD、8RU-M5
39RU-GEN1	39RU-GEN1、39RU-M5
39RU-M5	39RU-GEN1、39RU-M5

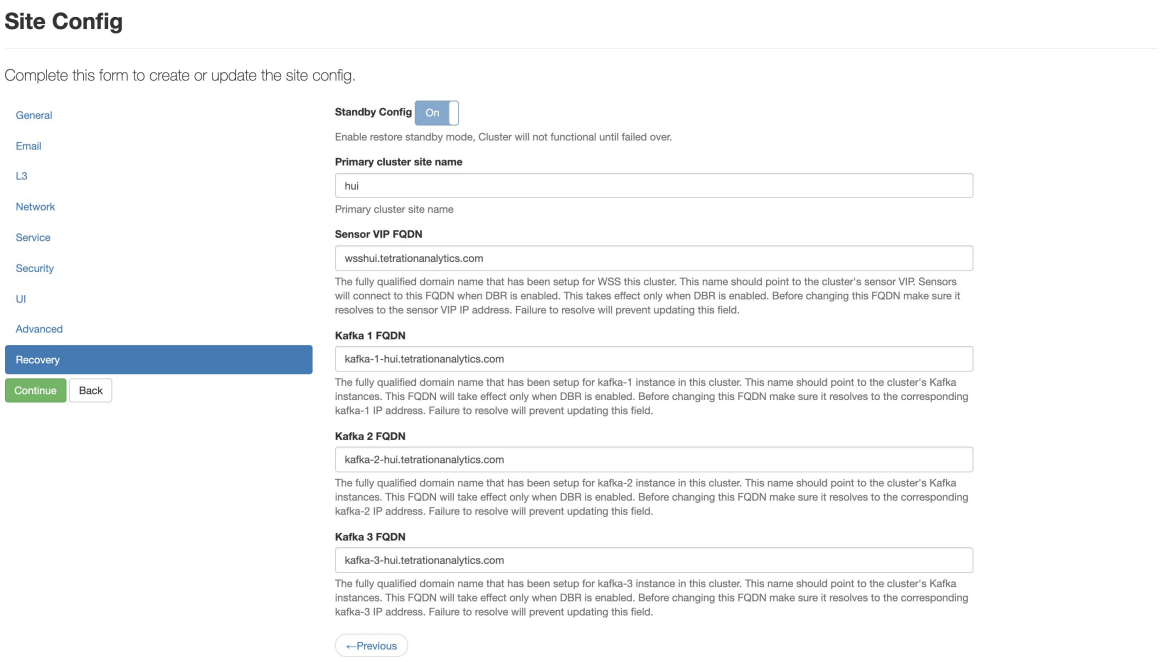
スタンバイモードでのクラスタの展開

データの復元を開始するには、シスコに連絡してください。

サイト情報でリカバリオプションを設定することにより、クラスタをスタンバイモードで展開できます。展開中にサイト情報を設定するときに、展開中の設定UIの[リカバリ（Recovery）]タブで復元の詳細を設定します。

スタンバイを展開するには3つのモードがあり、3つすべてのモードで次の設定を行います。

- [スタンバイ設定（Standby Config）]を[オン（On）]に設定します。この設定は、一度設定するとクラスタが再展開されるまで変更できません。
- プライマリクラスタ名とFQDNを設定します。この設定は、後で変更できます。



展開の残りの部分は、Cisco Secure Workloadクラスタの通常の展開と同じです。

クラスタがスタンバイモードになると、Cisco Secure WorkloadのUIにバナーが表示されます。

展開後にプライマリクラスタ名とFQDNを再設定して、スタンバイクラスタが別のクラスタを追跡できるようにすることが可能です。この設定は、[クラスタ設定（Cluster Configuration）]ページからフェールオーバーがトリガーされる前に、後で再設定できます。

スタンバイ展開モード

- **コールドスタンバイ**：スタンバイクラスタはありません。ただし、プライマリクラスタはデータをS3にバックアップします。災害時には、新しいクラスタ（またはプライマリと同じクラスタ）をプロビジョニングし、スタンバイモードで展開して復元する必要があります。
- **ウォームスタンバイ**：スタンバイクラスタが使用可能であり、スタンバイモードで展開されています。災害発生時に使用できるように、S3クラスタから定期的に状態を取得し、準備完了状態にします。災害時には、この新しいクラスタにログインして、フェールオーバーをトリガーします。

- **ルークウォームスタンバイ**：複数のプライマリクラスタが、より少ないスタンバイクラスタによってバックアップされます。スタンバイクラスタはスタンバイモードで展開されます。災害発生後にのみ、ストレージバケット情報が設定され、データがプリフェッチされて、クラスタが復元されます。

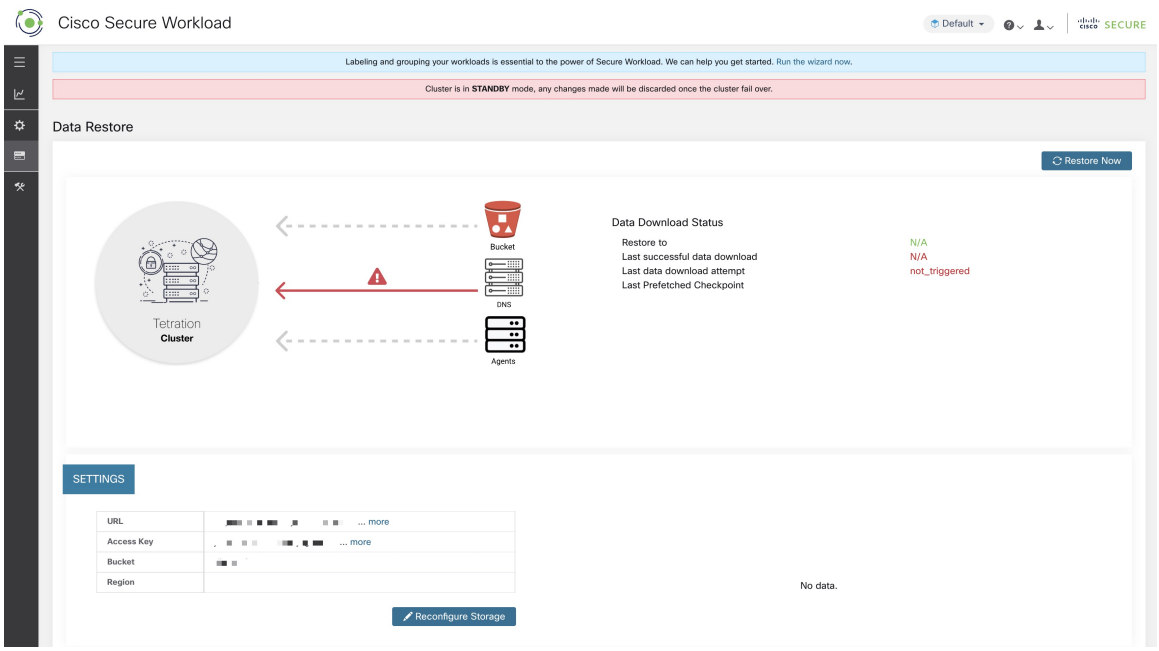
クラスタデータのプリフェッチ

クラスタを復元する前に、データをプリフェッチする必要があります。チェックポイントデータは、データのバックアップに使用されるのと同じストレージバケットからプリフェッチされます。バックアップサービスがストレージからダウンロードできるようにするには、ログイン情報を提供する必要があります。ストレージがプリフェッチ用に設定されていない場合、[データの復元（Data Restore）] タブによりセットアップウィザードが起動されます。

（注） スタンバイクラスタは、S3 ストレージとのみ対話します。プライマリクラスタのバックアップを更新して別のストレージまたはバケットを使用する場合、スタンバイクラスタのストレージを更新する必要があります。

情報が検証されると、ストレージはプリフェッチ用に自動設定されます。復元タブにプリフェッチステータスが表示されます。

図 *456* : プリフェッチステータス



ステータスページには、次の詳細が表示されます。

- 左上のセクションには、復元を開始するためのさまざまなコンポーネントの準備ができていることを示す図があります。データを確認するには、コンポーネントにカーソルを合わせます。関連データが右上のセクションに表示されます。

- [バケット（Bucket）]：プリフェッチステータスが表示されます。最新のデータが 45 分以上前のものである場合、赤で表示されます。アクティブでのバックアップが各チェックポイントで 45 分以上かかる場合、最新のデータが 45 分以上古いことは問題ではないことに注意してください。
- [DNS]：スタンバイクラスタ IP アドレスに関する Kafka および WSS FQDN 解決が表示されます。復元中に、FQDN がスタンバイクラスタ IP アドレスに更新されない場合、エージェントは接続できません。FQDN がスタンバイクラスタへの解決を開始した後に、ステータスが緑色に変わります。
- [エージェント（Agents）]：スタンバイクラスタに正常に切り替えられたソフトウェアエージェントの数が表示されます。これは復元がトリガーされた後にのみ関係します。

- 右上のセクションには、左側のセクションで選択したグラフィックに関連する情報が表示されます。[今すぐ復元（Restore Now）] をクリックすると、復元プロセスが開始されます。
- 左下のセクションには、使用中のプリフェッチストレージ設定が表示されます。
- 右下のセクションには、プリフェッチ遅延のグラフが表示されます。

データのプリフェッチは、迅速な復元を確実にするために、いくつかの必要なコンポーネントを更新します。データのプリフェッチを完了できない場合は、失敗の理由がステータスページに表示されます。

プリフェッチの失敗原因となる一般的なエラー：

S3 アクセスエラー：この場合、ストレージからのデータを正常にダウンロードできませんでした。これは、無効なログイン情報、ストレージポリシーの変更、または一時的なネットワークの問題が原因で発生する可能性があります。

互換性のないクラスタバージョン：プライマリクラスタと同じバージョン（同じパッチバージョンを含む）の Cisco Secure Workload を実行しているクラスタにデータを復元できます。これは、アップグレード中にクラスタの 1 つだけがアップグレードされている場合に発生することがあります。または、展開中に、別のバージョンが展開に使用されている場合です。クラスタを共通バージョンに展開すると、この問題が解消されます。

互換性のない SKU バージョン：プライマリクラスタについて、スタンバイクラスタで許可されている SKU をメモします。プライマリクラスタの SKU の復元には、特定の SKU のみが許可されます。

クラスタの復元

クラスタの復元は、[復元ステータス（Restore Status）] ページの右上のセクションにある [今すぐ復元（Restore Now）] をクリックしてトリガーできます。復元アクションがトリガーされる前に、確認を求められます。

クラスタデータは、次の 2 つのフェーズで復元されます。

- 必須フェーズ：サービスを再開するために必要なデータが最初に復元されます。必須フェーズにかかる時間は、設定、インストールされているソフトウェアエージェントの数、バックアップされているデータの量、およびフローメタデータによって異なります。必須フェーズ中は、UI にアクセスできません。**必須フェーズに UI にアクセスする必要が生じた場合、サポートを受けるにはワーキング TA ゲストキーが必要です。**
- レイジーフェーズ：クラスタデータ（フローデータを含む）はバックグラウンドで復元され、クラスタの使用はブロックされません。クラスタ UI にアクセスでき、復元の完了率を示すバナーが表示されます。このフェーズ中、クラスタは動作可能であり、データパイプラインは正常に機能し、フロー検索も使用できます。

目標復旧時間と目標復旧時点

この項では、データのバックアップおよび復元ソリューションの目標復旧時間（RTO）と目標復旧時点（RPO）について説明します。

プライマリクラスタで開始されたバックアップは、バックアップされるデータの量とバックアップ設定に応じて、完了するまでに時間がかかります。さまざまなバックアップモードによって、ソリューションの RPO が定義されます。

- スケジュールされている場合、継続的でないバックアップが使用され、バックアップは 1 日に 1 回開始されます。災害が発生した場合、失われるデータの最大時間は、約 24 時間 + バックアップストレージにデータをコピーするためにかかる時間です。そのため、RPO は少なくとも 24 時間です。
- 継続的なモードのバックアップを使用する場合は、新しいバックアップが前のバックアップの 15 分後に開始されます。各バックアップで、作成のために一定の時間がかかり、その後、データをバックアップストレージにアップロードするために一定の時間がかかります。最初のバックアップは完全バックアップで、その後のバックアップは増分バックアップです。増分バックアップにはそれほど時間がかかりません。災害が発生した場合、失われるデータの量は、バックアップの作成にかかった時間と、ストレージへのバックアップのアップロードにかかった時間の合計になります。通常、この場合の RPO は約数分から 1 時間です。

クラスタを復元する場合、まず必須データがストレージからプリフェッチされ、次に必須復元フェーズがトリガーされます。必須復元フェーズ中は、UI を使用できません。必須復元が完了すると、UI を使用できるようになります。残りのデータは、遅延復元フェーズで復元されます。この場合の RTO は、必須フェーズが完了した後、UI が使用可能になるまでにかかる時間です。RTO は、スタンバイ展開モードによって異なります。

- コールドスタンバイモード：このモードでは、クラスタを最初に展開する必要があります。これには約数時間かかります。次に、バックアップストレージのログイン情報を使用してクラスタを設定する必要があります。初めてバックアップがスタンバイクラスタにアップロードされるため、取得して処理する必要がある必須データが多数あります。プリフェッチにかかる時間は約数十分です（バックアップされるデータの量によって異なります）。必須復元フェーズが完了するまで約 30 分かかります。まとめると、主にクラスタの起動と展開にかかる時間が原因で、RTO 時間は約数時間になります。

- ルーク ウォーム スタンバイ モード：このモードでは、クラスタはすでに展開されていますが、バックアップストレージは設定されていません。バックアップストレージのログイン情報を使用してクラスタを設定する必要があります。初めてバックアップがスタンバイクラスタにアップロードされるため、取得して処理する必要がある必須データが多数あります。プリフェッチにかかる時間は約数十分です（バックアップされるデータの量によって異なります）。必須復元フェーズが完了するまで約30分かかります。まとめると、バックアップされるデータの量とバックアップストレージからデータをプルするためにかかる時間に応じて、RTO 時間は約 1 ～ 2 時間になります。
- ウォームスタンバイモード：このモードでは、クラスタがすでに展開され、バックアップストレージが設定されていて、プリフェッチによりストレージからデータが取得されています。クラスタはすぐに復元できるようになります。復元により必須復元フェーズがトリガーされ、完了するまで約 30 分かかります。RTO 時間は約 30 分になります。アクティブからストレージにバックアップがアップロードされてから、スタンバイによってバックアップがプルされるまでには、ある程度の遅延が生じることに注意してください。これには約数分かかります。（ディザスタイベントが発生する前に）アクティブから最新のバックアップがスタンバイにプリフェッチされていない場合は、取得されるまで数分間待つ必要があります。

データのバックアップと復元を行うアップグレード

クラスタでデータのバックアップと復元が有効になっている場合は、アップグレードを開始する前にスケジュールを非アクティブ化することを推奨します。「バックアップスケジュールの非アクティブ化」を参照してください。これにより、アップグレードが開始される前に正常なバックアップが存在し、新しいバックアップがアップロードされないことが保証されます。失敗したチェックポイントが作成されることを回避するため、スケジュールの非アクティブ化は、チェックポイントが進行中でないときに実行する必要があります。

トラブルシューティング：データのバックアップと復元

S3 設定チェックの失敗

ストレージテストが失敗した場合は、右側のペインに表示される障害シナリオを特定し、次のことを確認します。

- S3 準拠のストレージ URL が正しい
- ストレージのアクセスキーと秘密鍵が正しい
- ストレージ上にバケットが存在し、正しいアクセス（読み取り/書き込み）権限が付与されている
- ストレージに直接アクセスする必要がある場合は、プロキシが設定されている
- Cohesity を使用している場合は、マルチパート アップロード オプションが無効になっている

S3 設定チェックのエラーシナリオ

次の表は、一般的なエラーシナリオと解決策を示したものであり、すべてを網羅したものではありません。

表 *37 : S3* 設定チェック中のエラーメッセージと解決策

エラー メッセージ	シナリオ	対処法
検索不可（Not found）	正しくないバケット名	ストレージに設定されているバケットの正しい名前を入力します
SSL接続エラー（SSL connection error）	SSL 証明書の有効期限または検証のエラー	SSL 証明書を確認します
	無効な HTTPS URL	• ストレージの正しい HTTPS URL を再入力します。 • SSL 証明書の検証中に発生した障害を解決します。
接続のタイムアウト	S3 サーバーの IP アドレスに到達できません	クラスタと S3 サーバーの間のネットワーク接続を確認します
URLに接続できません（Unable to connect to URL）	正しくないバケットリージョン	正しいバケットのリージョンを入力します
	無効な URL	S3 ストレージエンドポイントの正しい URL を再入力します
Forbidden	無効な秘密鍵	ストレージの正しい秘密鍵を入力します
	無効なアクセスキー	ストレージの正しいアクセスキーを入力します
S3設定を確認できません（Unable to verify S3 configuration）	その他の例外または一般的なエラー	しばらくしてから S3 ストレージの設定を試みます

チェックポイントのエラーコード

次の表は、チェックポイントの一般的なエラーコードを示したものであり、すべてを網羅したものではありません。

表 *38*：チェックポイントのエラーコード

エラー コード	説明
E101：DBのチェックポイントの失敗（E101: DB checkpoint failure）	Mongodb oplog のスナップショットを取得できません
E102：フローデータのチェックポイントの失敗（E102: Flow data checkpoint failure）	Druidデータベースのスナップショットを取得できません
E103：DBスナップショットのアップロードの失敗（E103: DB snapshot upload failure）	Mongo DB スナップショットをアップロードできません
E201：DBのコピーの失敗（E201: DB copy failure）	Mongo スナップショットを HDFS にアップロードできません
E202：設定のコピーの失敗（E202: Config copy failure）	Consul-Vault スナップショットを HDFS にアップロードできません
E203：設定のチェックポイントの失敗（E203: Config checkpoint failure）	consul-vault データのチェックポイントを実行できません
E204：チェックポイント中の設定データの不一致（E204: Config data mismatch during checkpoint）	最大再試行回数後に consul/vault チェックポイントを生成できません
E301：バックアップデータのアップロードの失敗（E301: Backup data upload failure）	HDFS チェックポイントの失敗
E302：チェックポイントのアップロードの失敗（E302: Checkpoint upload failure）	Copydriver が S3 にデータをアップロードできませんでした
E401：チェックポイント中のシステムアップグレード（E401: System upgrade during checkpoint）	このチェックポイント中にクラスタがアップグレードされました。チェックポイントは使用できません
E402：チェックポイント中のサービスの再起動（E402: Service restart during checkpoint）	Bkpdriver が作成状態で再起動しました。チェックポイントは使用できません
E403：前のチェックポイントの失敗（E403: Previous checkpoint failure）	前回の実行でチェックポイントが失敗しました
E404：別のチェックポイントが進行中（E404: Another checkpoint in progress）	別のチェックポイントが進行中です
E405：チェックポイントを作成できない（E405: Unable to create checkpoint）	チェックポイントのサブプロセスでエラーが発生しました

エラー コード	説明
失敗：完了（Failed: Completed）	先行するチェックポイントの一部が失敗しました。同時に開始する複数のチェックポイントが重複している可能性があります

VM の情報

[トラブルシューティング（Troubleshoot）] メニューの [仮想マシン（Virtual Machine）] ページには、Cisco Secure Workload クラスタの一部であるすべての仮想マシンが表示されます。クラスタの起動またはアップグレード（あれば）中の展開ステータス、さらにパブリック IP も表示されます。クラスタ内のすべての VM はパブリックネットワークの一部ではないため、パブリック IP を持たない場合があることに注意してください。

クラスタのアップグレード

アップグレードオプションにアクセスするには、左側のナビゲーションバーで [プラットフォーム（Platform）] > [アップグレード/再起動/シャットダウン（Upgrade/Reboot/Shutdown）] をクリックします。

アップグレードには 2 種類があります。ここでは、「フル」アップグレードプロセスについて説明します。このアップグレード中に、Orchestrator-VM を除くクラスタ内のすべての VM がシャットダウンされ、新しい VM が展開され、サービスが再プロビジョニングされます。クラスタ内のすべてのデータは、このアップグレード中に保持されます。ただし、このアップグレード中に発生する約 2 時間のダウンタイムを除きます。

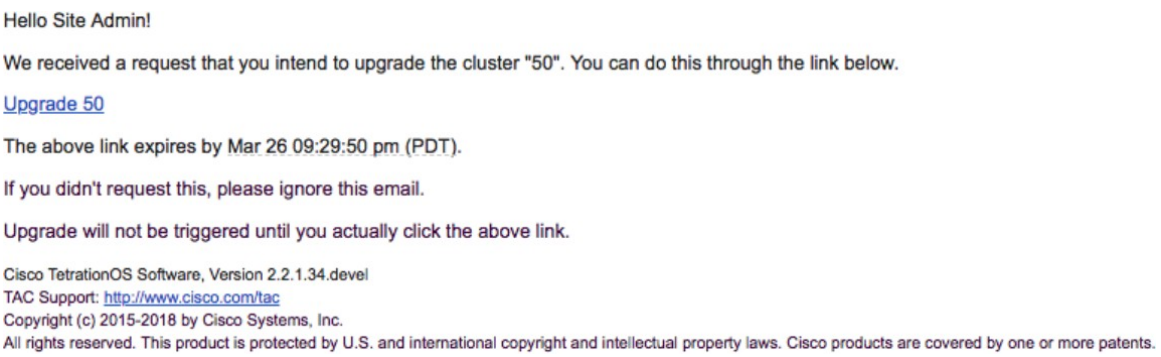
アップグレードの開始

アップグレードを開始するには、左側のナビゲーションバーで [プラットフォーム（Platform）] > [アップグレード/再起動/シャットダウン（Upgrade/Reboot/Shutdown）] をクリックします。

アップグレードページには、クラスタのアップグレード/パッチアップグレード/シャットダウン/再起動オプションがあります。

フルアップグレードを開始するには、[アップグレードリンクの送信（Send Upgrade Link）] をクリックします。フルアップグレードを実行すると、オーケストレータ VM 以外のすべての VM がシャットダウンされ、それらすべてがアップグレードおよび再展開されます。このため、2 時間以上のクラスタのダウンタイムが発生します。パッチアップグレードを実行するとダウンタイムは最小限に抑えられますが、パッチの適用が必要なサービスが更新されるだけで、VM は再起動されません。ダウンタイムは通常、数分程度です。パッチアップグレードを開始するには、[パッチアップグレードリンクの送信（Send Patch Upgrade Link）] をクリックします。電源を切った後にクラスタの再起動を開始するには、[再起動リンクの送信（Send Reboot Link）] を使用します。これらのリンクのいずれかをクリックすると、リンクを含む電子メールが生成され、アップグレードを開始したユーザーに送信されます。

図 *457 :* フルアップグレードの開始


Hello Site Admin!

We received a request that you intend to upgrade the cluster "50". You can do this through the link below.

Upgrade 50

The above link expires by Mar 26 09:29:50 pm (PDT).

If you didn't request this, please ignore this email.

Upgrade will not be triggered until you actually click the above link.

Cisco TetrationOS Software, Version 2.2.1.34.devel
TAC Support: http://www.cisco.com/tac
Copyright (c) 2015-2018 by Cisco Systems, Inc.
All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. Cisco products are covered by one or more patents.

オーケストレータは電子メールを送信する前に、いくつかの検証チェックを実行して、クラスタがアップグレード可能であることを確認します。検証チェックの内容は次のとおりです。

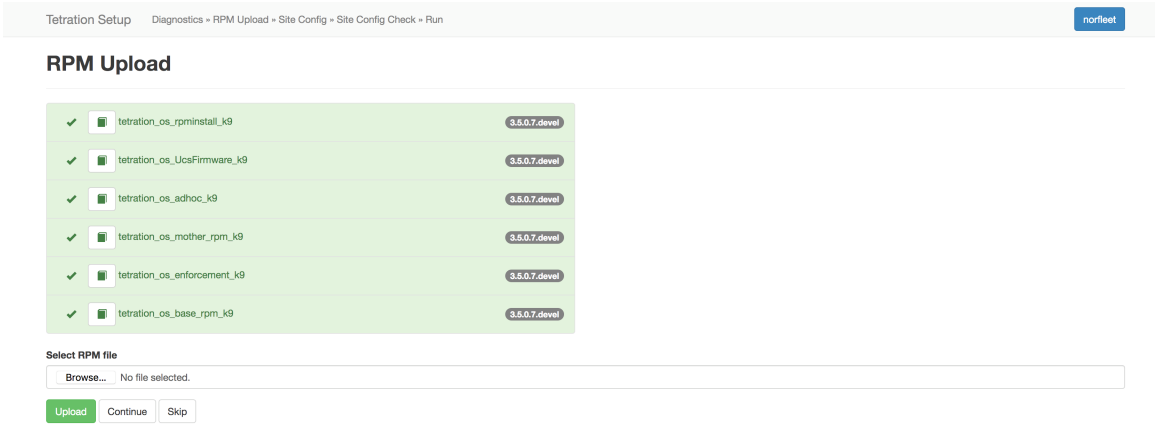
1. 稼働停止中のノードがないことを確認します。
2. 各ベアメタルをチェックして、ハードウェア障害がないことを確認します。ハードウェア障害には以下が含まれます。
    1. ドライブの障害
    2. ドライブの予測可能な障害
    3. ドライブの欠落
    4. StorCLI の障害
    5. MCE ログエラー
3. すべての BM が稼働状態であることを確認します。39RU の場合はサーバーが 36 台以上、8RU の場合は 6 台以上であることを確認します。

いずれかの障害がある場合は、アップグレードリンクは送信されません。HWエラーやホスト欠落などの情報を含む500エラーが表示されるため、オーケストレータログで詳細を確認します。このシナリオでは、ホストの orchestrator.service.consul にある /local/logs/tetration/orchestrator/orchestrator.log で、最後の 100 個のエラーメッセージを確認できます。ここで、3 つのチェックポイントのどれが障害の原因であるかに関する詳細情報が提供されます。このとき、通常はハードウェアの修正とノードの再稼働が必要になります。それが完了したら、[アップグレードリンクの送信（Send Upgrade Link）] をクリックしてアップグレードを再開できます。

RPM アップロード

電子メールのリンクをクリックすると、クラスタのセットアップUIに接続します。セットアップUIは、クラスタの展開とアップグレードで使用する操作UIです。最初のページには、現在クラスタにインストールされている RPM のリストが表示されます。このページは、すべての RPM をアップロードするためのアップロードページでもあります。

図 *458 : RPM* アップロード



セットアップ UI に表示される順序で RPM をアップロードします。順序は次のとおりです。

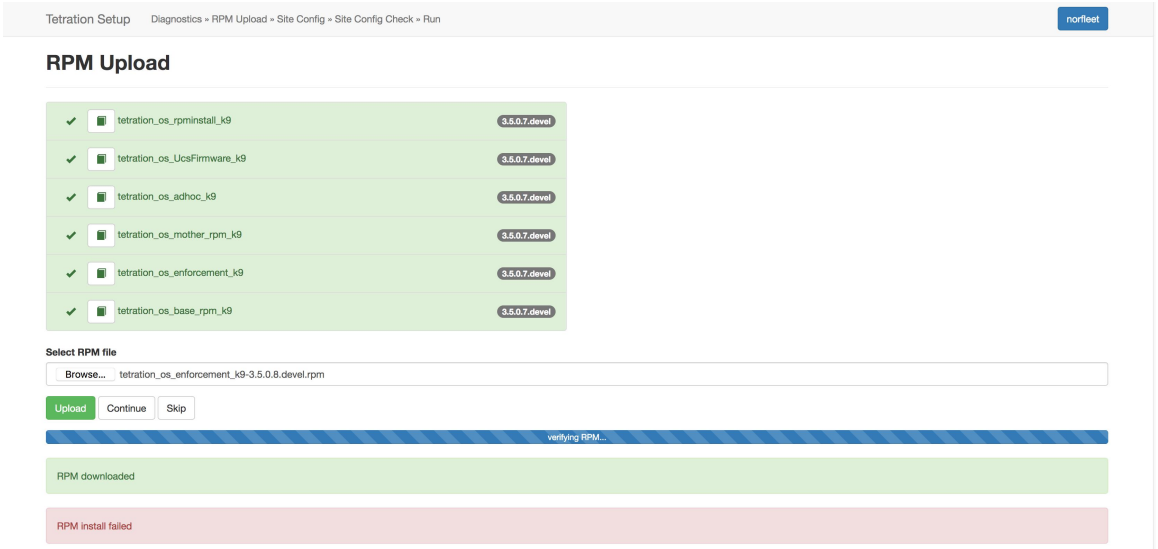
1. tetration_os_rpminstall_k9
2. tetration_os_UcsFirmware_k9
3. tetration_os_adhoc_k9
4. tetration_os_mother_rpm_k9
5. tetration_os_enforcement_k9
6. tetration_os_base_rpm_k9

（注） vSphere に展開された Secure Workload 仮想クラスタの場合は、必ず tetration_os_ova_k9 RPM もアップグレードしてください。tetration_os_base_rpm_k9 はアップグレードしないでください。

これ以外の順序でアップロードすると、アップロードが失敗します。すべての RPM が正しい順序でアップロードされるまで、[続行（Continue）] ボタンは無効になります。

各アップロードのログは、それぞれの RPM の左側にあるログ記号をクリックして表示できます。また、失敗したアップロードは赤色でマークされます。

図 *459 : RPM* アップロードログ



サイト情報

次のステップは、サイト情報を更新することです。すべてのサイト情報フィールドが更新可能というわけではありません。次のフィールドのみを更新できます。

1. SSH 公開キー
2. Sentinel アラート電子メール（Bosun 用）
3. CIMC 内部ネットワーク
4. CIMC 内部ネットワークゲートウェイ
5. 外部ネットワーク。注：既存の外部ネットワークは変更しないでください。既存のネットワークに付加することで、さらにネットワークを追加できます。既存のネットワークを変更または削除すると、クラスタが使用できなくなります。
6. DNS リゾルバ
7. DNS ドメイン
8. NTP サーバ
9. SMTP サーバー（SMTP Server）
10. SMTP ポート（SMTP Port）
11. SMTP ユーザー名（オプション）
12. SMTP パスワード（オプション）
13. Syslog サーバー（オプション）

14. Syslog ポート（オプション）
15. Syslog シビラティ（重大度）（オプション）



（注） Syslogサーバーのシビラティ（重大度）は、クリティカルから情報提供までの範囲です。Bosunアラートのシビラティ（重大度）は、警告以上（情報提供）に設定する必要があります。



（注） バージョン 3.1 以降、セットアップ UI を介した外部 syslog はサポートされていません。ユーザーは、syslog にデータをエクスポートするように TAN アプライアンスを設定する必要があります。詳細については、「外部 syslog トンネリングの TAN への移行」を参照してください。



（注） Secure Workload は、STARTTLS コマンドを介した SSL/TLS 通信を行うメールサーバーとの安全なSMTP通信をサポートします。安全なトラフィックをサポートするサーバーの標準ポートは、通常は 587/TCP ですが、多くのサーバーは標準の 25/TCP ポートでも安全な通信を受け入れます。

Secure Workload は、外部メールサーバーと通信するための SMTPS プロトコルをサポートしていません。

残りのフィールドは更新できません。変更がない場合は、[続行（Continue）] をクリックしてアップグレード前のチェックをトリガーします。変更がある場合は、フィールドを更新して [続行（Continue）] をクリックします。

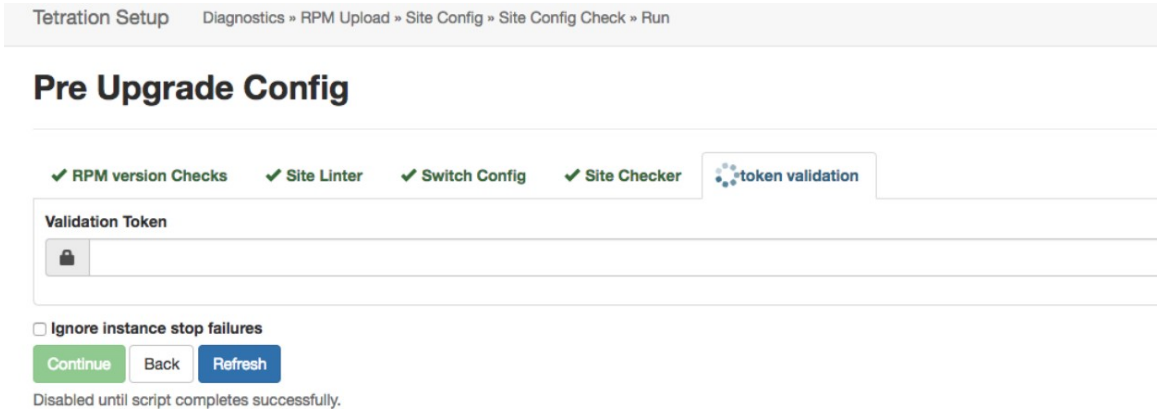
アップグレードの事前チェック

アップグレードを開始する前に、クラスタでいくつかのチェックを行い、アップグレードを開始する前に問題がないことを確認します。

1. RPM バージョンチェック：すべての RPM がアップロードされ、バージョンが正しいことを確認します。状態が正しいかどうかを確認するのではなく、アップロードされたかどうかを確認するだけです。状態チェックは、アップロード自体の一部として実行されます。
2. サイトリンター：サイト情報のリンティングを行います。
3. スイッチ構成：リーフ/スパインスイッチを構成します。
4. サイトチェッカー：DNS、NTP、および SMTP サーバーのチェックを行います。最後にトークン付きの電子メールを送信します。このメールは、プライマリサイトの管理者アカウントに送信されます。DNS、NTP、または SMTP のいずれかのサービスが使用できない場合、この手順は失敗します。

5. トークンの検証：電子メールで送信されたトークンを入力し、[続行（Continue）]をクリックします。

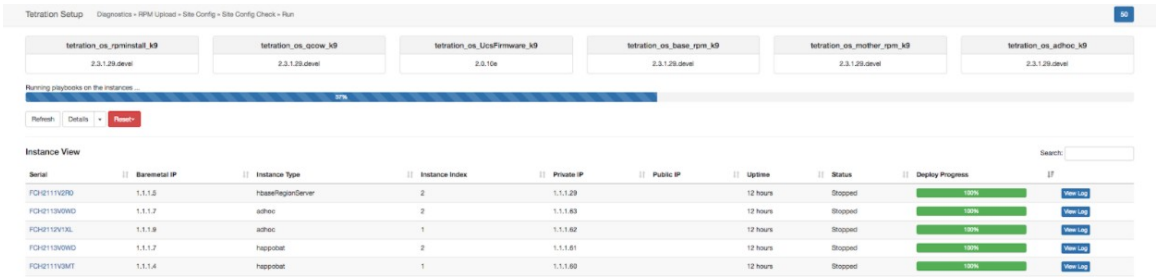
図 460 : アップグレードの事前チェック



クラスタのアップグレード

アップグレード前の手順が完了したら、「トークンの確認メール」で受け取ったトークンを入力した後に、[続行（Continue）]をクリックすると、アップグレードを開始できます。[障害時に停止を無視（Ignore Stop Failures）] という追加オプションがありますが、このオプションをオンにしてはいけません。これは、特定のサービスがシャットダウンせず、アップグレードが失敗した場合の回復オプションです。このオプションを使用すると、VM が強制的にシャットダウンされ、サービスの再稼働時に障害が発生する可能性があります。このオプションは、エンジニアの監督下で使用してください。

図 461 : クラスタのアップグレード



[続行（Continue）] をクリックすると、アップグレードが開始されます。

手順

ステップ 1 右上のクラスタ名をクリックすると、使用されているサイト情報が表示されます。

ステップ 2 その下には、すべての Tetration_os RPM とそのバージョンがあります。

ステップ 3 グローバル アップグレード バーには、アップグレードの進行状況が表示されます。進行中は青色、完了時は緑色、失敗時は赤色になります。進行状況バーのすぐ上に、アップグレードの現在のステータスが表示されます。

ステップ 4 また、次の 3 つのボタンがあります。

a) [更新（Refresh）] : ページを更新します

b) [詳細]（Details）] : クリックすると、このアップグレード中に完了したすべてのステップが表示されます。ステップの横にある矢印をクリックすると、利用可能なすべてのログが表示されます。詳細については別途記します。

c) [リセット（Reset）] : オーケストレータの状態をリセットするオプションです。このオプションを選択すると、アップグレードがキャンセルされて、最初に戻ります。アップグレードに失敗した場合を除き、このボタンを使用しないでください。また、アップグレードが失敗した後、アップグレード再開前にすべてのプロセスが完了するまで数分かかります。

d) [再開（Resume）] : アップグレードに失敗すると、失敗した段階に応じて、再開オプションが表示されます。[再開（Resume）] をクリックすると、前回の安定していた部分からアップグレードが再開されます。

ステップ 5 次に、インスタンスビューが表示されます。個々のVMの展開ステータスがすべて追跡されます。インスタンスビューは次の列で構成されます。

a) [シリアル（Serial）] : この VM をホストするベアメタルのシリアル番号

b) [ベアメタルIP（Baremetal IP）] : このベアメタルに割り当てられた内部 IP

c) [インスタンスタイプ（Instance Type）] : VM のタイプ

d) [インスタンスインデックス（Instance Index）] : VM のインデックス - 高可用性向けに同じタイプの VM が複数あります。

e) [プライベートIP（Private IP）] : この VM に割り当てられた内部 IP

f) [パブリックIP（Pubic IP）] : この VM に割り当てられたルーティング可能な IP - すべての VM にあるわけではありません。

g) [稼働時間（Uptime）] : VM の稼働時間

h) [ステータス（Status）] : [停止（Stopped）]、[展開（展開済み）]、[失敗（Failed）]、[未開始（未開始）]、[進行中（In Progress）] のいずれかです。

i) [展開の進行状況（Deploy Progress）] : 展開が完了した割合

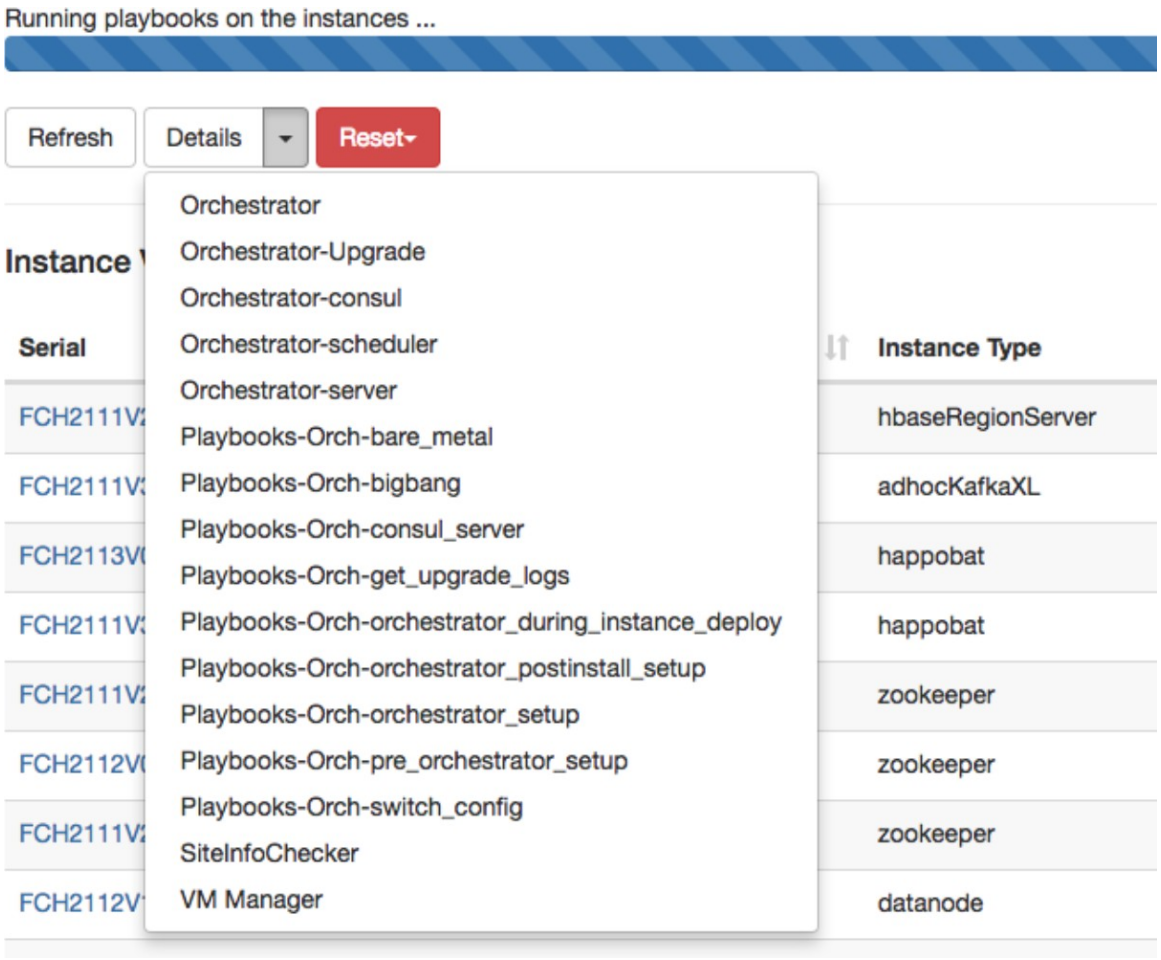
j) [ログの表示（View Log ）] : VM の展開ステータスを表示するためのボタン

ログ

ログには 2 つのタイプがあります。

1. VM 展開ログ：これらのログは、[ログの表示（View Log）] ボタンをクリックして表示できます。
2. オーケストレーションログ。詳細ボタンの横にある矢印をクリックすると、これらが表示されます。次のように表示されます。

図 *462* : ログ



各リンクはログを指します。

手順

ステップ 1 [Orchestrator] - オーケストレータログ：これは進行状況を追跡する最初の場所です。エラーが発生すると、別のログを指して参照します。

ステップ 2 [Orchestrator-Upgrade] - 2.3 の NOP

ステップ 3 [Orchestrator-consul]：プライマリオーケストレータで実行される consul ログ

ステップ 4 [Orchestrator-Scheduler] - VM スケジューラログ：どの VM がどのベアメタルに配置されたかを示すログと、スケジューリングログ

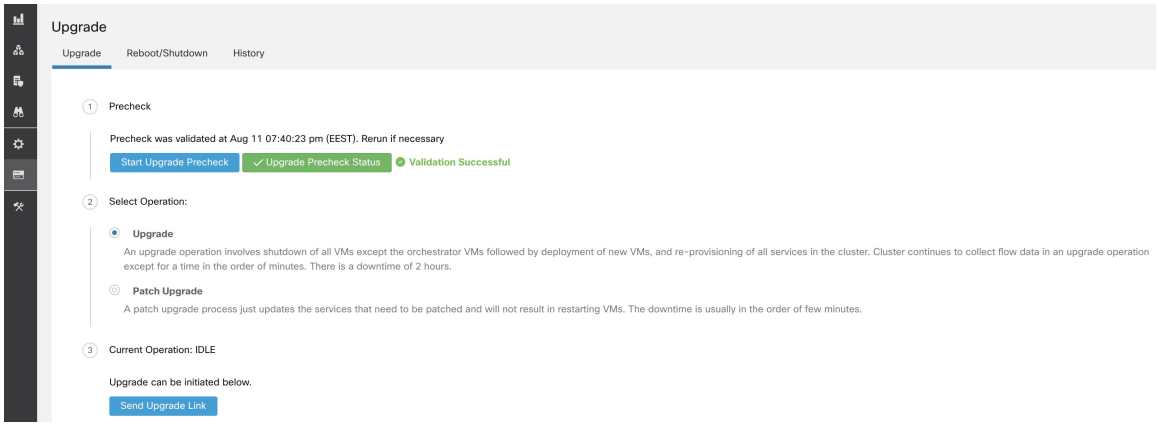
ステップ 5 [Orchestrator-server]：オーケストレータからの HTTP サーバーログ

ステップ 6 [Playbooks-*]：オーケストレータで実行されるすべての playbook ログ。

アップグレードの事前チェックをいつでも実行

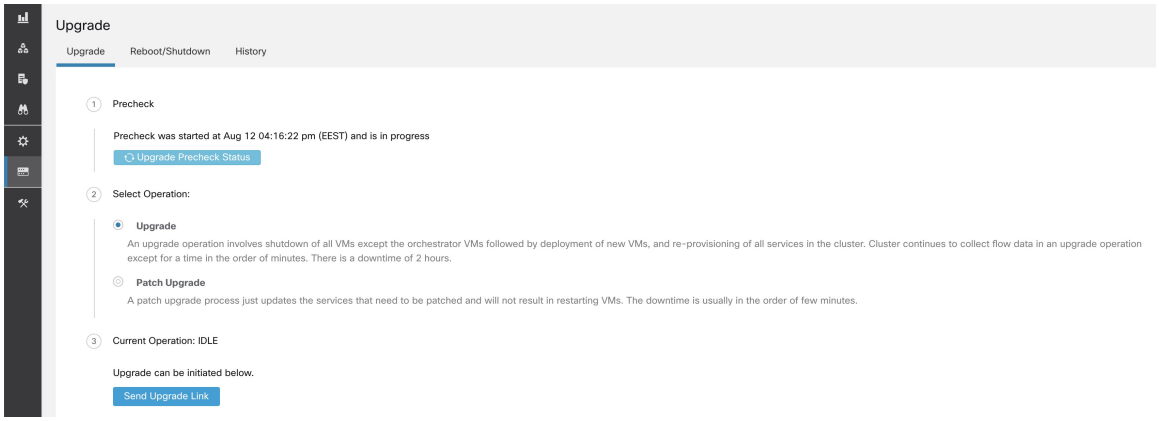
場合によっては、アップグレードをスケジュールした後、アップグレードを開始しているときに、ハードウェア障害が発生するか、クラスタをアップグレードする準備ができていないことがあります。アップグレードを続行する前に、これを修正する必要があります。アップグレードウィンドウまで待つ代わりに、アップグレードの事前チェックをいつでも開始できます。これらのチェックは、アップグレード/パッチ/再起動の開始時を除き、いつでも何度でも実行できます。アップグレードの事前チェックを任意のタイミングで実行するには、[アップグレード（Upgrade）] ページに移動します。

図 *463*：アップグレードの事前チェックをいつでも実行する手順



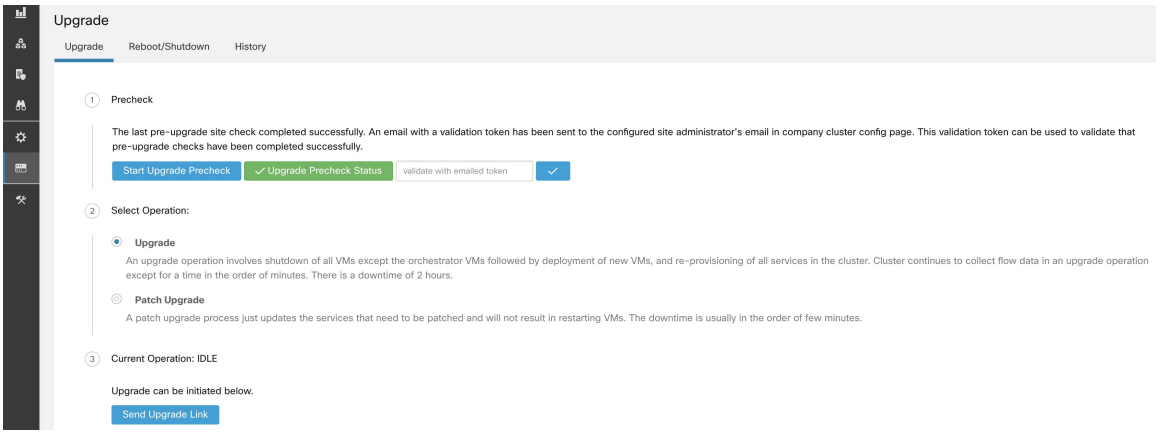
[アップグレードの事前チェックの開始（Start Upgrade Precheck）] をクリックします。これにより、アップグレードの事前チェックが開始され、実行状態に移行します。

図 *464*：アップグレードの事前チェックをいつでも実行する手順



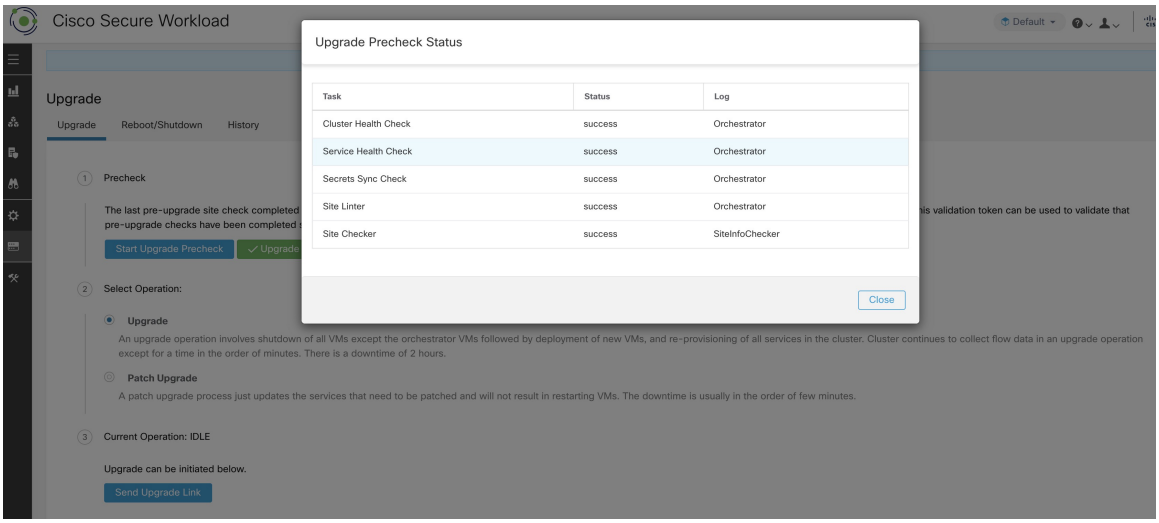
この間、オーケストレータはすべてのアップグレードの事前チェックを実行します。すべてのチェックに合格すると、チェックを開始したユーザーに、電子メールトークンが記載された電子メールが送信されます。トークンを入力して、アップグレードの事前チェックを完了します。

図 *465 :* アップグレードの事前チェックをいつでも実行する手順



アップグレードの事前チェック中にエラーが発生した場合、失敗状態に移行し、失敗したタスクが表示されます。ステータスはいつでも確認でき、新しいダイアログボックスに表示されます。

図 *466 :* アップグレードの事前チェックをいつでも実行する手順



Data Backup and Restore（DBR）

クラスタで **DBR** が有効になっている場合は、「データのバックアップと復元を行うアップグレード」も参照してください。

スナップショット

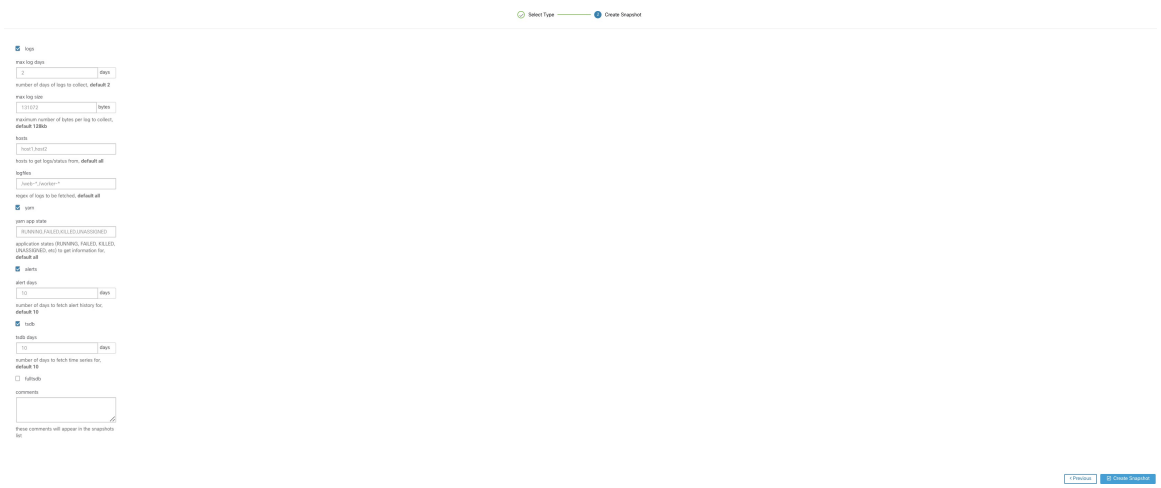
スナップショット作成ユーザーインターフェイスへのアクセス

カスタマーサポートロールを持つユーザーは、ウィンドウの左側にあるナビゲーションバーから **[トラブルシューティング（Troubleshoot）]** > **[スナップショット（Snapshots）]** を選択して、スナップショットツールにアクセスできます。

スナップショットツールを使用して、クラシックスナップショットまたは Cisco Integrated Management Controller（CIMC）テクニカルサポートバンドルを作成できます。スナップショット ファイル リスト ページで [スナップショットの作成（Create Snapshot）] ボタンをクリックすると、クラシックスナップショットまたはCIMC スナップショット（テクニカルサポートバンドル）を選択するページがロードされます。CIMC スナップショットを選択するオプションは、Secure Workload ソフトウェアのみ（ESXi）および Secure Workload SaaS では無効になっています。

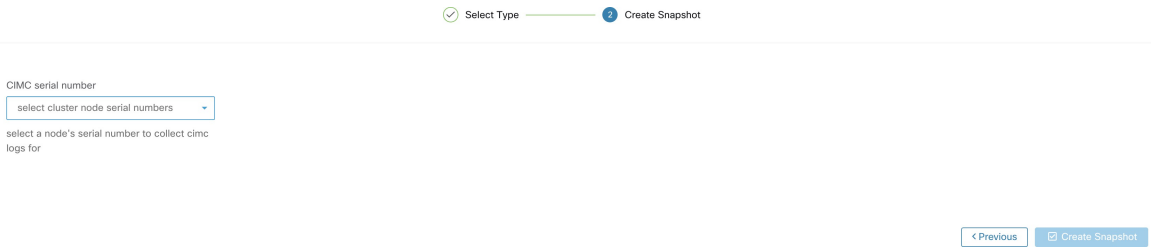
[クラシックスナップショット（Classic Snapshot）] ボタンをクリックすると、スナップショットツールを実行するためのユーザーインターフェイスがロードされます。

図 *467 :* スナップショットツールの実行



[CIMC スナップショット（CIMC Snapshot）] ボタンをクリックすると、CIMC テクニカルサポートツールを実行するためのユーザーインターフェイスがロードされます。

図 *468 : CIMC* テクニカルサポートの実行



スナップショットの作成

[スナップショットの作成（Create Snapshot）] でデフォルトのオプションを選択すると、スナップショットツールは次の情報を収集します。

- ログ
- Hadoop/YARN アプリケーションの状態とログ
- アラート履歴
- さまざまな TSDB 統計情報

デフォルトをオーバーライドして、特定のオプションを指定することができます。

- ログオプション
  - 最大ログ日数（max log days）：収集するログの日数、デフォルトは 2。
  - 最大ログサイズ（max log size ）：収集するログごとの最大バイト数、デフォルトは 128 KB。
  - ホスト（hosts）：ログ/ステータスを取得するホスト、デフォルトは [すべて（all）]。
  - ログファイル（logfiles）：取得するログの正規表現、デフォルトは [すべて（all）]。
- yarn オプション
  - yarn アプリの状態（yarn app state）：情報を取得するアプリケーションの状態（[実行中（RUNNING）]、[失敗（FAILED）]、[強制終了（KILLED）]、[未割り当て（UNASSIGNED）] など）。デフォルトは all。
- アラートオプション
  - アラート日数（alert days）：アラートデータを収集する日数。
- tsdb オプション
  - tsdb 日数（tsdb days）：tsdb データを収集する日数。この値を増やすと、非常に大規模なスナップショットが作成される可能性があります。

- fulltsdb オプション
  - fulltsdb：startTime、endTime fullDumpPath、localDumpFile、nameFilterIncludeRegex を指定し、収集するメトリックを制限するために使用できる JSON オブジェクト。
- コメント（comments）：スナップショットを収集する理由や収集するユーザーを記載するために追加できます。

[スナップショットの作成（Create Snapshot）] を選択すると、スナップショット ファイル リストページの上部にスナップショットの進行状況バーが表示されます。スナップショットが完了したら、スナップショット ファイル リスト ページの [ダウンロード（Download）] ボタンを使用してダウンロードできます。一度に収集できるスナップショットは 1 つだけです。

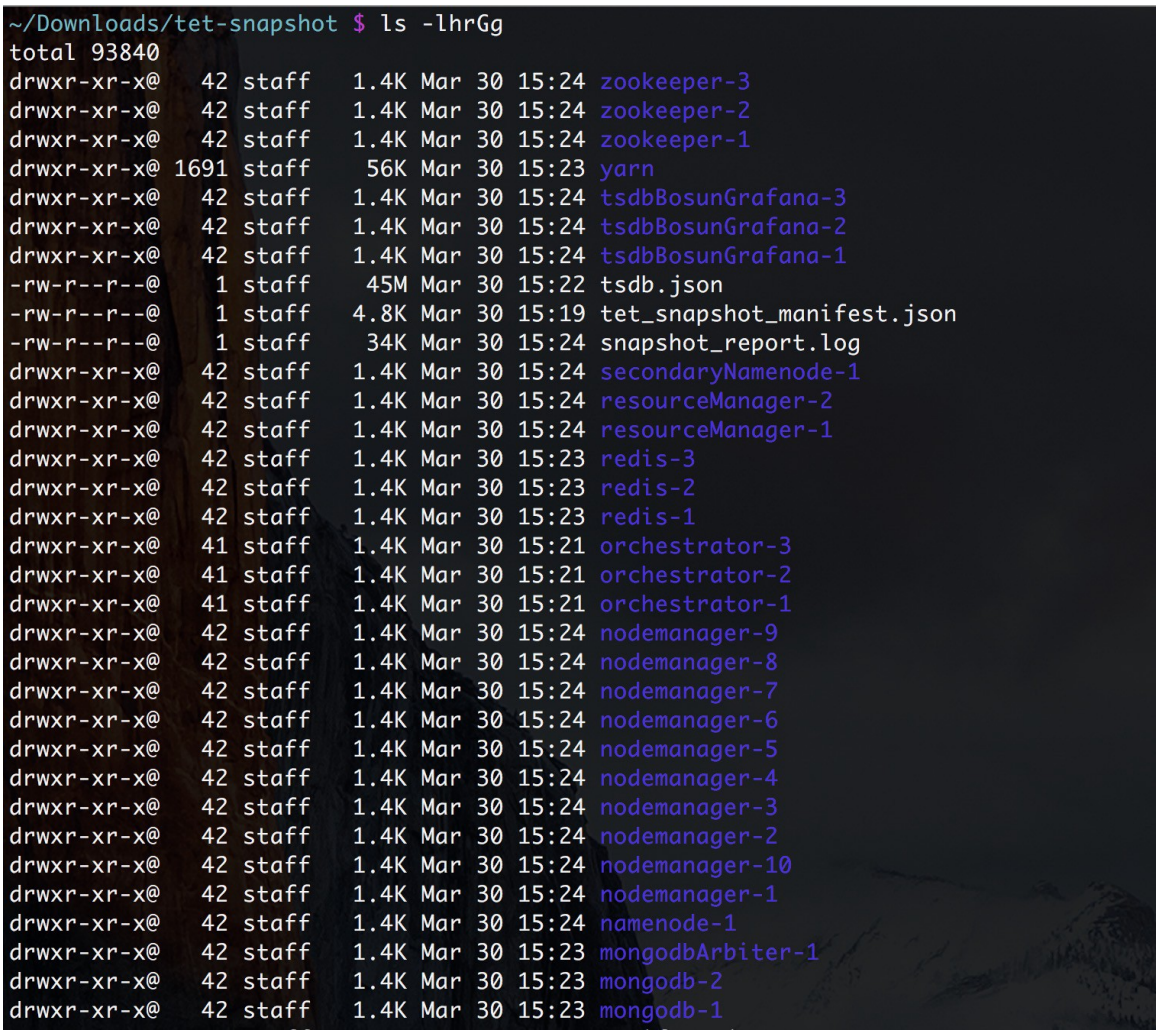
CIMC テクニカルサポートバンドルの作成

CIMC スナップショット（テクニカルサポートバンドル）ページで、CIMC テクニカルサポートバンドルを作成するノードのシリアル番号を選択し、[スナップショットの作成（Create Snapshot）] ボタンをクリックします。CIMC テクニカルサポートバンドル収集の進捗バーがスナップショット ファイル リスト ページに表示され、コメントセクションには CIMC テクニカルサポートバンドル収集がトリガーされたことが反映されます。CIMC テクニカルサポートバンドルの収集が完了すると、スナップショット ファイル リスト ページからファイルをダウンロードできます。

スナップショットの使用

スナップショットを解凍すると、各マシンのログを含む ./clustername_snapshot ディレクトリが作成されます。ログは、マシンのいくつかのディレクトリのデータを含むテキストファイルとして保存されます。スナップショットは、JSON 形式でキャプチャされたすべての Hadoop/TSDB データも保存します。

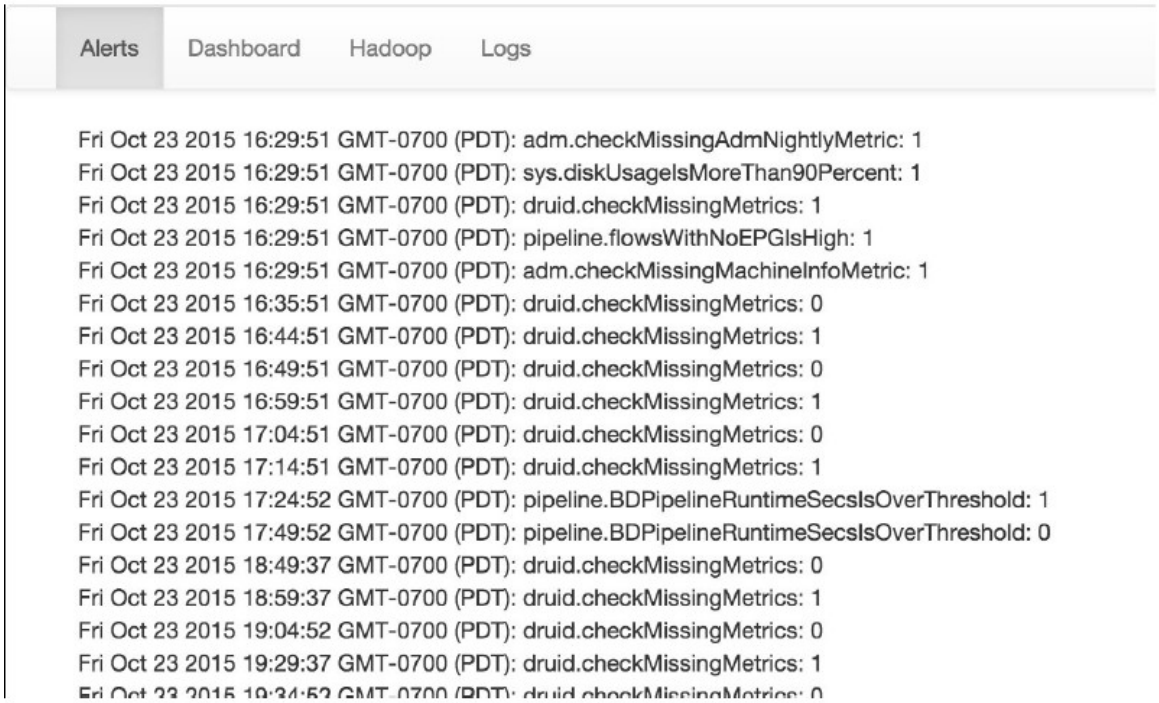
図 *469* : スナップショットの使用



パッケージ化された index.html をブラウザで開くと、次のタブが表示されます。

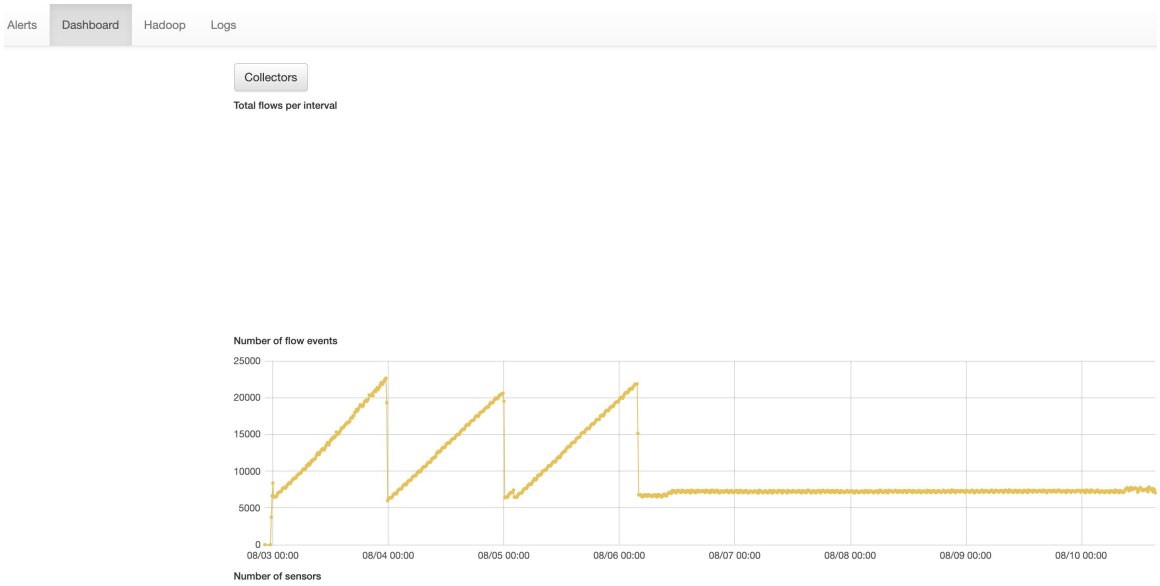
- アラート状態の変化についての簡潔なリスト。

図 *470 :* アラート状態の変化についての簡潔なリスト



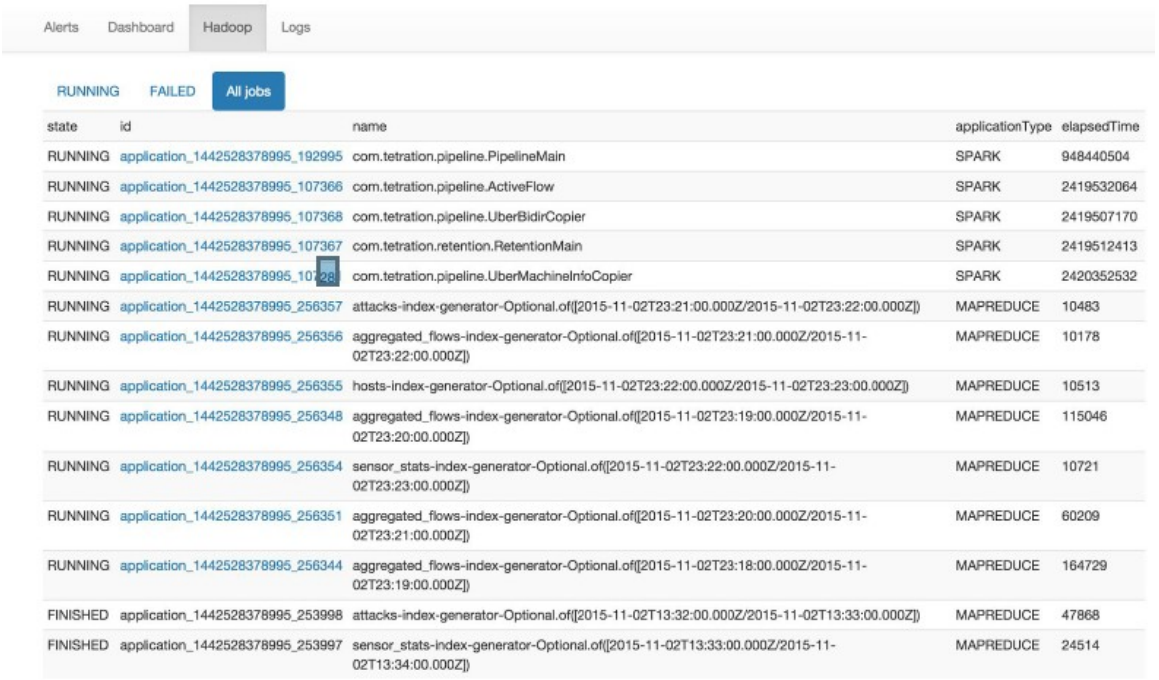
- grafana ダッシュボードの複製。

図 *471 : grafana* ダッシュボードの複製



- ジョブとその状態を含む Hadoop Resource Manager のフロントエンドの複製。ジョブを選択すると、そのジョブのログが表示されます。

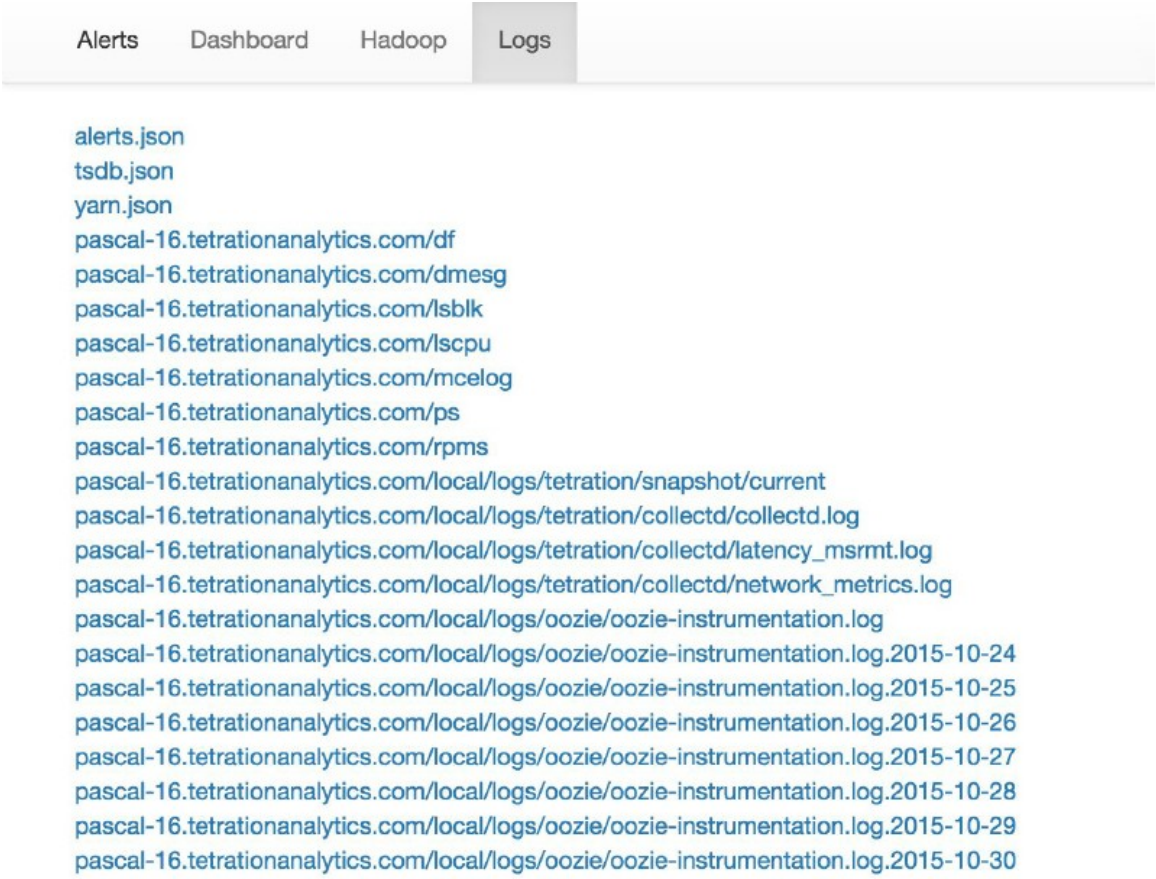
図 *472 : Hadoop Resource Manager* の複製



state	id	name	applicationType	elapsedTime
RUNNING	application_1442528378995_192995	com.tetration.pipeline.PipelineMain	SPARK	948440504
RUNNING	application_1442528378995_107366	com.tetration.pipeline.ActiveFlow	SPARK	2419532064
RUNNING	application_1442528378995_107368	com.tetration.pipeline.UberBidirCopier	SPARK	2419507170
RUNNING	application_1442528378995_107367	com.tetration.retention.RetentionMain	SPARK	2419512413
RUNNING	application_1442528378995_10736[illegible]	com.tetration.pipeline.UberMachineInfoCopier	SPARK	2420352532
RUNNING	application_1442528378995_256357	attacks-index-generator-Optional.of([2015-11-02T23:21:00.000Z/2015-11-02T23:22:00.000Z])	MAPREDUCE	10483
RUNNING	application_1442528378995_256356	aggregated_flows-index-generator-Optional.of([2015-11-02T23:21:00.000Z/2015-11-02T23:22:00.000Z])	MAPREDUCE	10178
RUNNING	application_1442528378995_256355	hosts-index-generator-Optional.of([2015-11-02T23:22:00.000Z/2015-11-02T23:23:00.000Z])	MAPREDUCE	10513
RUNNING	application_1442528378995_256348	aggregated_flows-index-generator-Optional.of([2015-11-02T23:19:00.000Z/2015-11-02T23:20:00.000Z])	MAPREDUCE	115046
RUNNING	application_1442528378995_256354	sensor_stats-index-generator-Optional.of([2015-11-02T23:22:00.000Z/2015-11-02T23:23:00.000Z])	MAPREDUCE	10721
RUNNING	application_1442528378995_256351	aggregated_flows-index-generator-Optional.of([2015-11-02T23:20:00.000Z/2015-11-02T23:21:00.000Z])	MAPREDUCE	60209
RUNNING	application_1442528378995_256344	aggregated_flows-index-generator-Optional.of([2015-11-02T23:18:00.000Z/2015-11-02T23:19:00.000Z])	MAPREDUCE	164729
FINISHED	application_1442528378995_253998	attacks-index-generator-Optional.of([2015-11-02T13:32:00.000Z/2015-11-02T13:33:00.000Z])	MAPREDUCE	47868
FINISHED	application_1442528378995_253997	sensor_stats-index-generator-Optional.of([2015-11-02T13:33:00.000Z/2015-11-02T13:34:00.000Z])	MAPREDUCE	24514

- 収集されたすべてのログのリスト

図 *473* : 収集されたすべてのログのリスト

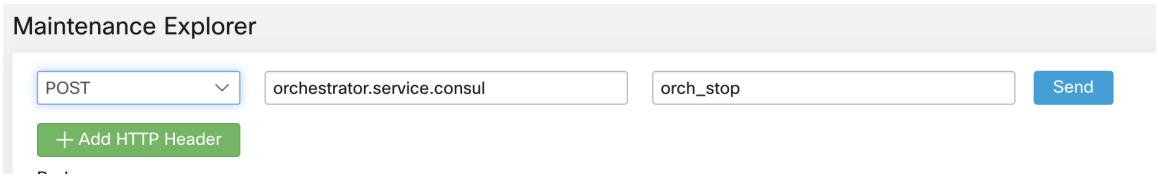


デバッグとメンテナンスにスナップショットサービスを使用

スナップショットサービスを使用してサービスコマンドを実行できますが、これにはカスタマーサポート権限が必要です。

Explore ツール （**[トラブルシュート（Troubleshoot）] > [メンテナンスエクスプローラ（Maintenance Explorer)]**）を使用すると、クラスタ内の任意の URI に到達できます。

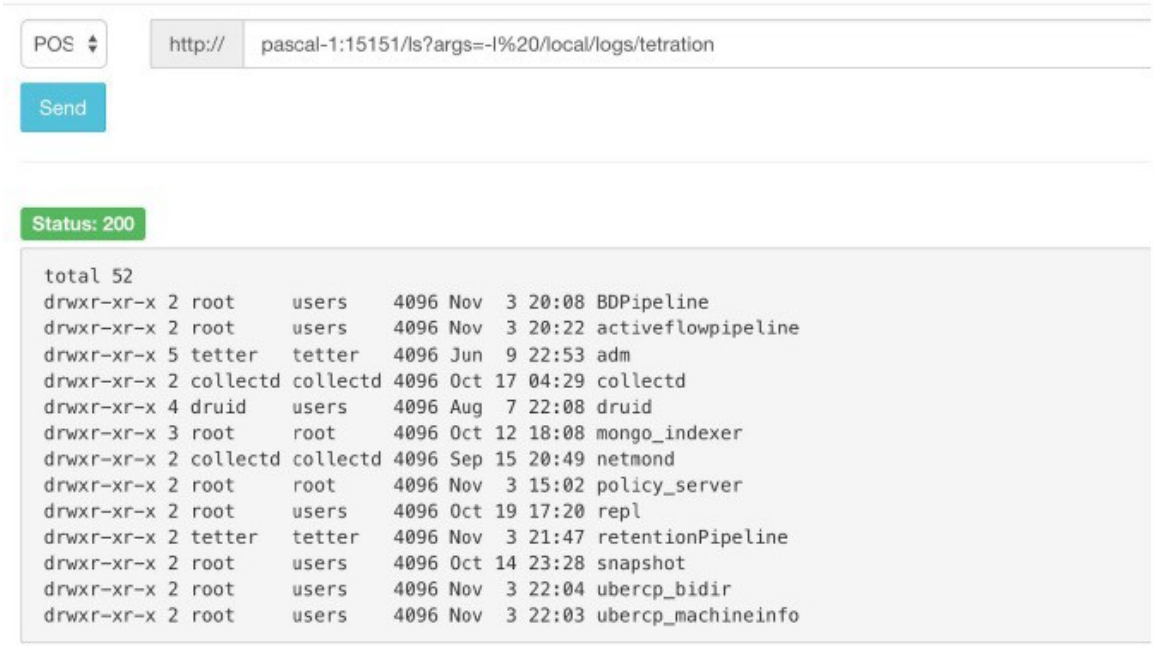
図 *474* : デバッグとメンテナンスにスナップショットサービスを使用する例



Explore ツールは、カスタマーサポートの権限を持つユーザーのみに表示されます。

スナップショットサービスは、すべてのノードのポート 15151 で実行されます。内部ネットワークのみでリッスンし（外部には公開されません）、さまざまなコマンド用の POST エンドポイントがあります。

図 *475*: デバッグとメンテナンスにスナップショットサービスを使用する例



到達する必要がある URI は **POST** http://<hostname>:15151/<cmd>?args=<args> です。ここで args はスペースで区切られ、URI はエンコードされます。この URI によってシェルでコマンドが実行されることはありません。これにより、何かが実行されることを回避できます。

スナップショットのエンドポイントは、次に対して定義されています。

- **snapshot 0.2.5**

  –ls

  –**svstatus、svrestart - sv status、sv restart を実行** 例：1.1.11.15:15151/svrestart?args=snapshot

  –hadoopls - hadoop fs -ls <args> を実行

  –hadoopdu - hadoop fs -du <args> を実行

  –**ps** 例：1.1.11.31:15151/ps?args=eafux

  –du

  –ambari - ambari_service.py を実行

  –monit

  –MegaCli64（/usr/bin/MegaCli64）

  –service

  –hadoopfsck - hadoop -fsck を実行

- **snapshot 0.2.6**

  −makecurrent - make -C /local/deploy-ansible current を実行

  −netstat

- **snapshot 0.2.7（uid “nobody” として実行）**

  −cat

  −head

  −tail

  −grep

  −ip -6 neighbor

  −ip address

  −ip neighbor

別のエンドポイント POST /runsigned があります。これは、Secure Workload により署名されたシェルスクリプトを実行します。これは、POST されたデータ上で gpg -d を実行します。署名に対して検証できる場合は、暗号化されたテキストをシェルで実行します。これは、Ansible セットアップの一部として各サーバーに公開鍵をインポートすること、および秘密鍵を安全に保つ必要があることを意味します。

ランブック

カスタマーサポートの権限を持つユーザーは、ウィンドウの左側にあるナビゲーションバーから **[トラブルシューティング（Troubleshoot）]** > **[メンテナンスエクスプローラ（Maintenance Explorer）]** を選択して、ランブックを使用できます。ドロップダウンメニューから [POST] を選択します。（そうしないと、コマンドの実行時に Page Not Found エラーが発生します。）

スナップショット REST エンドポイントを使用してサービスを再起動します。

- **druid: 1.1.11.17:15151/service?args=supervisord%20restart**

  — druid ホストは、すべて .17 から .24 までの IP です。.17、.18 はコーディネータ、.19 はインデクサ、.20 ～ .24 はブローカです。

- **hadoop パイプラインランチャ**：

  −1.1.11.25:15151/svrestart?args=activeflowpipeline

  −1.1.11.25:15151/svrestart?args=adm

  −1.1.11.25:15151/svrestart?args=batchmover_bidir

  −1.1.11.25:15151/svrestart?args=batchmover_machineinfo

  −1.1.11.25:15151/svrestart?args=BDPipeline

  −1.1.11.25:15151/svrestart?args=mongo_indexer

  −1.1.11.25:15151/svrestart?args=retentionPipeline

- **ポリシー エンジン**

−1.1.11.25:15151/svrestart?args=policy_server

- **wss**

−1.1.11.47:15151/svrestart?args=wss

Explore / スナップショットのエンドポイントの概要

エンドポイントを実行するには、ウィンドウの左側にあるナビゲーションバーから **[トラブルシューティング（Troubleshoot）] > [メンテナンスエクスプローラ（Maintenance Explorer）]** ページに移動する必要があります。

また、**<end- point>?usage=true** のように任意のホストで **POST** コマンドを実行して、エクスプローラページで各エンドポイントの概要を表示することもできます。

例：**makecurrent?usage=true**

GET コマンド

エンドポイント	説明
bm_details	• ベアメタル情報を表示します。
endpoints	• ホスト上のすべてのエンドポイントを一覧表示します。
メンバー	• consul メンバーの現在のリストとそのステータスを表示します。
port2cimc	• ポートの接続先 IP を一覧表示します。 • **オーケストレータホスト**でのみ実行する必要があります。
status	• ホスト上のスナップショットサービスのステータスを表示します。
vm_info	• ロケーションの VM 情報を表示します。 • **ベアメタルホストのみ**で実行する必要があります。 • エンドポイントを **vm_info?args=<vmname>** として実行します。

POST コマンド

エンドポイント	説明
bm_shutdown_or_reboot	• 最初にそのホスト上のすべての仮想マシンをシャットダウンしてから、シャットダウンまたは再起動コマンドをベアメタルに発行することにより、ベアメタルホストのグレースフルシャットダウンまたはリブートを実行します。このエンドポイントを使用して、シャットダウンまたはリブートのステータスを取得することもできます。 • 任意のノードのシャットダウンまたはリブートステータスを取得するには、コマンド `bm_shutdown_or_reboot? query=serial=FCH2308V0FH` を使用します。 • ベアメタルのグレースフルシャットダウンを開始するには、`bm_shutdown_or_reboot? method=POST` を使用して、本文をホストのシリアル番号を表す JSON オブジェクトに設定します。例：`{"serial": "FCH2308V0FH"}` • ベアメタルのグレースフルリブートを開始するには、`bm_shutdown_or_reboot? method=POST` を使用して、本文をホストのシリアル番号を表す JSON オブジェクトに設定し、「true」に設定したリブートキーを含めます。例：`{"serial": "FCH2308V0FH", "reboot": true}`
cat	• UNIX 「cat」コマンドのラッパーコマンド
cimc_password_random	• CIMC パスワードをランダム化します。 • **オーケストレータホスト**でのみ実行する必要があります。
cleancmdlogs	• /lo- cal/logs/tetration/snapshot/cmdlogs/snapshot_cleancmdlogs_log 内のログをクリアします。

エンドポイント	説明
clear_sel	• システムイベントログをクリアします。 • **ベアメタルホストのみ**で実行する必要があります。
cluster_fw_upgrade	• これは**ベータ**機能です。 • クラスタ全体でUCSファームウェアアップグレードを実行します。 • これが正常に完了したら、各ベアメタルを再起動して、BIOSおよびその他のコンポーネント ファームウェアをアクティブ化する必要があります。 • **cluster_fw_upgrade** のように実行します。 • このエンドポイントは、ファームウェアのアップグレードを開始して監視し、アップグレードの段階が開始または完了したときにログファイルを更新します。 • 完全なアップグレードのステータスを取得するには、**cluster_fw_upgrade_status** エンドポイントを使用してください。
cluster_fw_upgrade_status	• これは**ベータ**機能です。 • 完全なクラスタUCSファームウェアアップグレードのステータスを取得します。 • **cluster_fw_upgrade_status** のように実行します。
cluster_powerdown	• クラスタの電源をオフにします。 • クラスタがダウンするため、注意して使用してください。 • エンドポイントを **cluster_powerdown?args=–start** のように実行します。
collector_status	• コレクタのステータスを表示します。 • **コレクタホストでのみ**実行する必要があります。

エンドポイント	説明
consul_kv_export	• consul からの k-v ペアを JSON 形式で表示します。 • **オーケストレータホスト**でのみ実行する必要があります。
consul_kv_recurse	• consul からの k-v ペアを表形式で表示します。 • **オーケストレータホスト**でのみ実行する必要があります。
df	• UNIX「df」コマンドのラッパーコマンド
dig	• UNIX「dig」コマンドのラッパーコマンド
dmesg	• UNIX「dmesg」コマンドのラッパーコマンド
dmidecode	• UNIX「dmidecode」コマンドのラッパーコマンド
druid_coordinator_v1	• DRUID のステータスを表示します。
du	• UNIX「du」コマンドのラッパーコマンド
dusorted	• UNIX「dusorted」コマンドのラッパーコマンド
externalize_change_tunnel	• CIMC UI をトンネリングするために使用されるコレクタ IP を変更します。 • **externalize_change_tunnel?method=POST** として実行します。 • 本文に **{“collector_ip” : “<IP>”}** を渡します。 • **オーケストレータホスト**でのみ実行する必要があります。

エンドポイント	説明
externalize_mgmt	• 各サーバーの CIMC UI の外部化の現在のステータスを表示します。 • 外部化のアドレスと残り時間を表示します。 • **オーケストレータホスト**でのみ実行する必要があります。
externalize_mgmt_read_only_password	• スイッチと CIMC UI の両方の読み取り専用パスワード（ta_guest）を変更します。 • 変更は、外部化された場合にのみ行われます。 • **externalize_mgmt_read_only_password?method=POST** のように実行します。 • **{"password" : "<password>"}** を本文に渡します。 • **オーケストレータホスト**でのみ実行する必要があります。
fsck	• UNIX「fsck」コマンドのラッパーコマンド • **ベアメタルホストのみ**で実行する必要があります。
get_cimc_techsupport	• BM の内部 IP アドレスを**入力**します。 • CIMCテクニカルサポートを取得します。 • 完了すると、UI のスナップショットページからダウンロードできるようになります。 • これは、クラスタ上の任意のホストから実行でき、引数としてベアメタルの内部 IP アドレスが必要です。 • 例：**get_cimc_techsupport?args=1.1.0.9**

エンドポイント	説明
syslog_endpoints	• 1 つ以上の UCS サーバーの syslog 構成を制御します。 • **パラメータの完全なリストを取得するには、-h を使用してコマンドを実行します。**
grep	• UNIX「grep」コマンドのラッパーコマンド
hadoopbalancer	• HDFS データをすべてのノードに均一に分散します。 • launcherhost などの **hdfs を持つホスト**で実行する必要があります。
hadoopdu	• hdfs のディレクトリ使用率を出力します。 • launcherhost などの **hdfs を持つホスト**で実行する必要があります。
hadoopfsck	• hadoop fsck を実行し、提供された hdfs ファイルシステムの状態を報告します。 • また、破損または欠落しているブロックをクリアする引数として「-delete」を使用します。 • 削除する前に、すべての DataNodes が起動していることを確認してください。そうしないと、データが失われる可能性があります。 • **ランチャホストでのみ**実行する必要があります。 • 状態をレポートするには、**hadoopfsck?args=/raw** のように実行します。 • 破損したファイルを削除するには、**hadoopfsck?args=/raw -delete** のように実行します。

エンドポイント	説明
hadoopls	• Hadoop ファイルシステムを一覧表示します。 • launcherhost などの **hdfs を持つホスト**で実行する必要があります。
hbasehbck	• 整合性およびテーブルの完全性の問題をチェックし、破損した HBase を修復します。 • **HBase ホストでのみ**実行する必要があります。 • 不整合を特定するには、**hbasehbck?args=-details** のように実行します。 • 破損した HBase を修復するには、**hbasehbck?args=-repair** のように実行します。 • **出力先 : /local/logs/tetration/snapshot/cmd_logs/snpsht_hbasehbck_log.txt** • **修復は、慎重に行ってください。**
hdfs_safe_state_recover	• HDFS を安全な状態から削除します。 • 容量に空きがなくスペースがクリアされているために HDFS が READ_ONLY_STATE の場合は必須です。 • **ランチャホストでのみ**実行する必要があります。 • **hadoopfs-rm'{{ hdfs_safe_state_marker_location }}/HDFS_READ_ONLY'** のように実行します。
initctl	• UNIX「initctl」コマンドのラッパーコマンド
head	• UNIX「head」コマンドのラッパーコマンド

エンドポイント	説明
internal_haproxy_status	• 内部 haproxy のステータスと統計を出力します。 • **オーケストレータホスト**でのみ実行する必要があります。
ip	• UNIX「ip」コマンドのラッパーコマンド
ipmifru	• Field Replaceable Unit（FRU、現場交換可能ユニット）の情報を出力します。 • **ベアメタルホストのみ**で実行する必要があります。
ipmilan	• LAN 構成を出力します。 • **ベアメタルホストのみ**で実行する必要があります。
ipmisel	• システムイベントログ（SEL）エントリを出力します。 • **ベアメタルホストのみ**で実行する必要があります。
ipmisensorlist	• IPMI センサー情報を出力します。 • **ベアメタルホストのみ**で実行する必要があります。
jstack	• 指定された Java プロセスまたはコアファイルの Java スレッドの Java スタックトレースを出力します。
ls	• UNIX「ls」コマンドのラッパー コマンド
lshw	• 「lshw」コマンドのラッパーコマンド
lsof	• UNIX「lsof」コマンドのラッパーコマンド
lvdisplay	• UNIX「lvdisplay」コマンドのラッパーコマンド

エンドポイント	説明
lvs	• UNIX「lvs」コマンドのラッパーコマンド
lvscan	• 「lvscan」コマンドのラッパーコマンド
makecurrent	• マーカーを現在のタイムスタンプに処理しているパイプラインをリセット/早送りします。 • **オーケストレータノードのみ**で実行する必要があります。 • エンドポイントを、**makecurrent?args=–start** のように実行します。
mongo_rs_status	• Mongo のレプリケーションステータスを表示します。 • **mongodb または enforcementpolicystore ホスト**のいずれかで実行する必要があります。
mongo_stats	• Mongo の統計情報を表示します。 • **mongodb または enforcementpolicystore ホスト**のいずれかで実行する必要があります。
mongodump	• データベースからコレクションをダンプします。 • **mongodb または enforcementpolicystore ホスト**のいずれかで実行する必要があります。 • mongodump?args=<collection>[–db DB] のように実行します。
monit	• UNIX「monit」コマンドのラッパーコマンド
namenode_jmx	• プライマリ namenode jmx メトリックを表示します。

エンドポイント	説明
ndisc6	• UNIX「ndisc6」コマンドのラッパーコマンド
netstat	• UNIX「netstat」コマンドのラッパーコマンド
ntpq	• UNIX「ntpq」コマンドのラッパーコマンド
orch_reset	• オーケストレータの状態をアイドルにリセットします。 • コミッショニングまたはデコミッショニングの失敗後に実行します。 • **orchestrator.service.consul ホストのみ**で実行する必要があります。 • **このコマンドを使用するときは、必ずカスタマーサポートに相談してください。**
orch_stop	• オーケストレータのプライマリを停止し、スイッチオーバーをトリガーします。 • **orchestrator.service.consul ホストのみ**で実行する必要があります。 • **慎重に使用してください。**
ping	• UNIX「ping」コマンドのラッパーコマンド
ping6	• UNIX「ping6」コマンドのラッパーコマンド
ps	• UNIX「ps」コマンドのラッパーコマンド
pv	• UNIX「pv」コマンドのラッパーコマンド
pvs	• UNIX「pvs」コマンドのラッパーコマンド
pvdisplay	• UNIX「pvdisplay」コマンドのラッパーコマンド

エンドポイント	説明
rdisc6	• UNIX「rdisc6」コマンドのラッパーコマンド
rebootnode	• ノードをリブートします。 • **ベアメタルホストのみ**で実行する必要があります。
recover_rpmdb	• ノード上の破損した RPMDB を回復します。 • ベアメタルまたは VM で実行可能です。
recoverhbase	• Hbase および TSDB サービスを回復します。 • **オーケストレータホスト**でのみ実行する必要があります。 • HDFS が正常なときに実行する必要があります。
recovervm	• stop/fsck/start を介して VM の回復を試みます。 • **オーケストレータホスト**でのみ実行する必要があります。 • エンドポイントを **recovervm?args=<vmname>** のように実行します。
restartservices	• すべての非UIサービスを停止して開始します。 • **orchestrator.service.consul ホストのみ**で実行する必要があります。 • **慎重に使用してください。** • エンドポイントを **restartservices?args=–start** のように実行します。

エンドポイント	説明
runsigned	• シスコが提供する署名付きスクリプトを実行します。 • **スクリプトガイドラインに記載されている手順に従ってください。**
service	• UNIX「service」コマンドのラッパーコマンド
smartctl	• smartctl 実行可能ファイルを実行します。 • ベアメタルノードのみで実行する必要があります。
storcli	• UNIX「storcli」コマンドのラッパーコマンド
sudocat	• /var/log または /local/logs でのみ機能する「cat」コマンドのラッパー
sudogrep	• /var/log または /local/logs のみで機能する「grep」コマンドのラッパー
sudohead	• /var/log または /local/logs のみで機能する「head」コマンドのラッパー
sudols	• 以下でのみ機能する「ls」コマンドのラッパー /var/log または /local/logs
sudotail	• 以下でのみ機能する「tail」コマンドのラッパー /var/log または /local/logs
sudozgrep	• /var/log または /local/logs のみで機能する「zgrep」コマンドのラッパー
sudozcat	• /var/log または /local/logs のみで機能する「zcat」コマンドのラッパー

エンドポイント	説明
svrestart	• 指定したサービスを再起動します。**svrestart?args=<servicename>** のようにコマンドを実行します。
svstatus	• 指定したサービスのステータスを出力します。**svstatus?args=<servicename>** のように実行します。
switchinfo	• クラスタスイッチに関する情報を取得します。
switch_namenode	• プライマリまたはセカンダリから namenode を手動でフェールオーバーします。 • **orchestrator.service.consul** ホストのみで実行する必要があります。 • namenode ホストのリコミッションまたはデコミッション中に実行します。 • エンドポイントを **switch_namenode?args=–start** のように実行します。
switch_secondarynamenode	• secondarynamenode をセカンダリからプライマリに手動でフェールオーバーします。 • **orchestrator.service.consul** ホストのみで実行する必要があります。 • namenode ホストのリコミッションまたはデコミッション中に実行します。 • エンドポイントを **switch_secondarynamenode?args=–start** のように実行します。

エンドポイント	説明
switch_yarn	• resourcemanager をプライマリからセカンダリへ、またはその逆に手動でフェールオーバーします。 • **orchestrator.service.consul ホストのみ**で実行する必要があります。 • sourcemanager ホストのリコミッションまたはデコミッション中に実行します。 • エンドポイントを **switch_yarn?args=–start** のように実行します。
tail	• UNIX「tail」コマンドのラッパーコマンド
toggle_chassis_locator	• ノードのシリアル番号で指定された物理ベアメタル上のシャーシロケータを切り替えます。 • 任意のノードから **toggle_chassis_locator?method=POST** のように実行します。 • 本文を、ホストのシリアル番号を記述する JSON オブジェクトに設定します（同時にサポートされるシリアル番号は 1 つだけです）。例：**{“serials”: [“FCH2308V0FH”]}**
tnp_agent_logs	• 外部オーケストレータとして登録されたロードバランサエージェントによって提供されるすべてのログ ファイルを使用してスナップショットを作成します。 • launcherhost ホストで実行する必要があります。

エンドポイント	説明
tnp_datastream	• 外部オーケストレータとして登録されたロードバランサポリシー適用エージェントによって消費されるポリシーストリームデータを使用してスナップショットを作成します。 • オーケストレータホストで実行する必要があります。 • ポリシー ステータス ストリーム データをダウンロードするには、エンドポイントを **tnp_datastream?args=–ds_type datasink** のように実行します。
ui_haproxy_status	• 外部 haproxy の haproxy 統計情報とステータスを出力します。
uptime	• UNIX「uptime」コマンドのラッパーコマンド
userapps_kill	• 実行中のすべてのユーザーアプリケーションを強制終了します。 • **launcherhost ホストのみ**で実行する必要があります。
vgdisplay	• UNIX「vgdisplay」コマンドのラッパーコマンド
vgs	• UNIX「vgs」コマンドのラッパーコマンド
vmfs	• VM 上のファイルシステムを一覧表示します。 • **ベアメタルホストのみ**で実行する必要があります。 • エンドポイントを **vmfs?args=<vmname>** のように実行します。

エンドポイント	説明
vminfo	• VM 情報を出力します。 • **ベアメタルホストのみ**で実行する必要があります。 • エンドポイントを **vminfo?args=<vmname>** のように実行します。
vmlist	• ベアメタル上のすべての VM を一覧表示します。 • **ベアメタルホストのみ**で実行する必要があります。 • エンドポイントを **vmlist?args=<vmname>** のように実行します。
vmreboot	• VM を再起動します。 • **ベアメタルホストのみ**で実行する必要があります。 • エンドポイントを **vmreboot?args=<vmname>** のように実行します。
vmshutdown	• VM のグレースフルシャットダウンを行います。 • **ベアメタルホストのみ**で実行する必要があります。 • エンドポイントを **vmshutdown?args=<vmname>** のように実行します。
vmstart	• VM を起動します。 • **ベアメタルホストのみ**で実行する必要があります。 • エンドポイントを **vmstart?args=<vmname>** のように実行します。

エンドポイント	説明
vmstop	• VM を強制的にシャットダウンします。 • **ベアメタルホストのみ**で実行する必要があります。 • エンドポイントを **vmstop?args=<vmname>** のように実行します。
yarnkill	• 実行中の Yarn アプリケーションを強制終了します。 • **launcherhost ホストのみ**で実行する必要があります。 • エンドポイントを **yarnkill?args=<application id>** のように実行します。 • すべてのアプリケーションを強制終了するには、**yarnkill?args=ALL** のように実行します。
yarnlogs	• Yarn アプリケーションログの最後の 500 MB をダンプします。 • **launcherhost ホストのみ**で実行する必要があります。 • エンドポイントを **yarnlogs?args=<application id> <job user>** のように実行します。
zcat	• UNIX「zcat」コマンドのラッパーコマンド
zgrep	• UNIX「zgrep」コマンドのラッパーコマンド

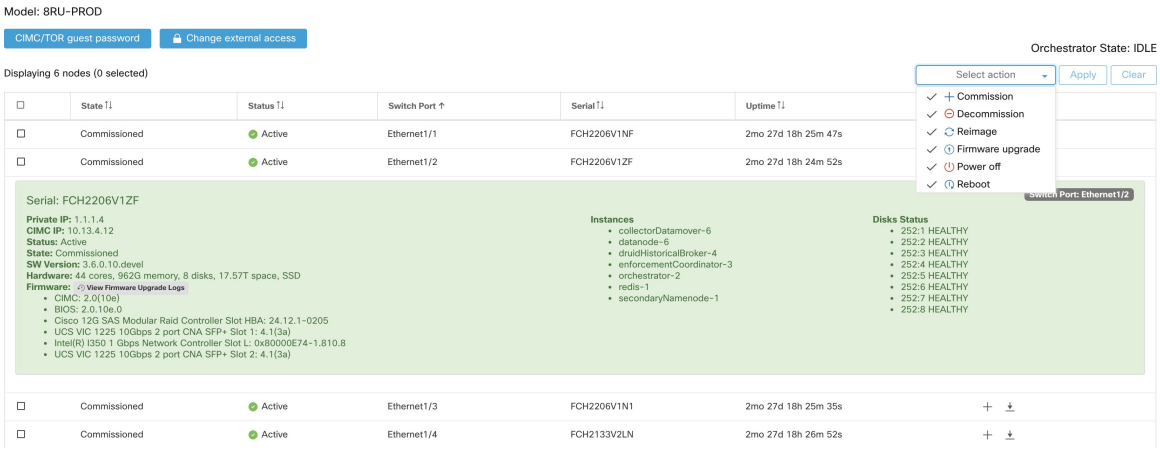
サーバーのメンテナンス

サーバーのメンテナンスには、故障したサーバーコンポーネント（ハードディスク、メモリなど）の交換、またはサーバー自体の交換が含まれます。

（注） クラスタ上にメンテナンスが必要なサーバーが複数ある場合は、一度に1つずつサーバーのメンテナンスを実行します。複数のサーバーを同時にデコミッションすると、データが失われる可能性があります。

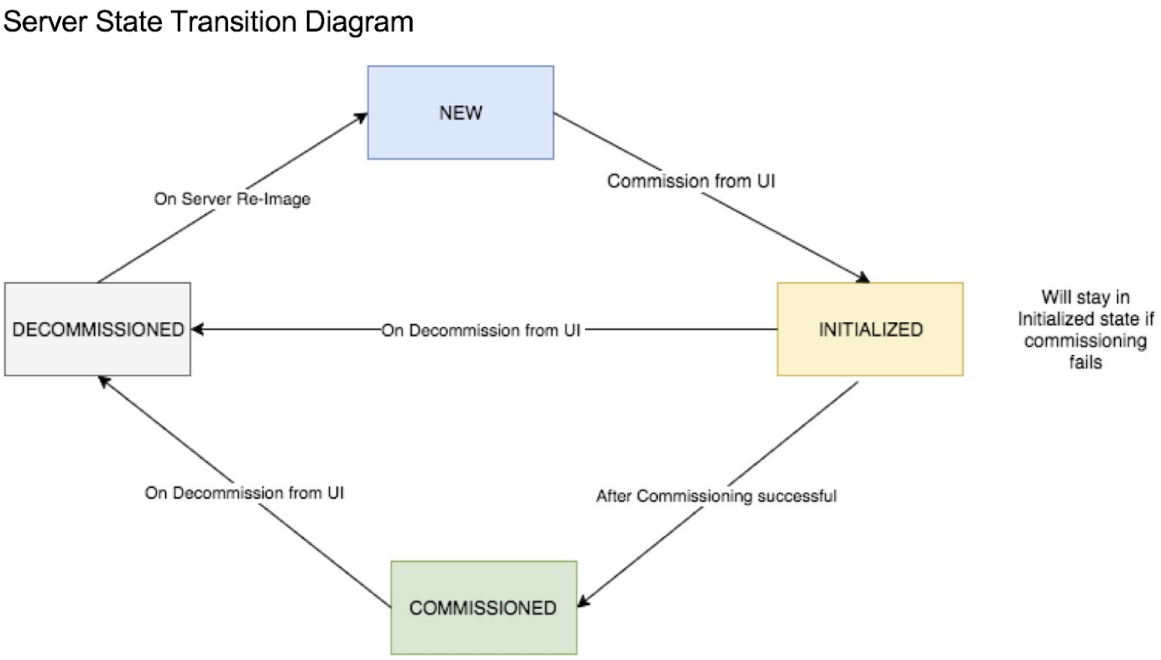
[クラスタステータス（Cluster Status）]ページ（左側のナビゲーションバーの[トラブルシュート（Troubleshoot）]メニューからアクセス）を使用して、サーバーのメンテナンスに関連するすべての手順を実行します。このページにはすべてのユーザーがアクセスできますが、アクションを実行できるのは**カスタマーサポート**ユーザーのみです。このページには、Cisco Secure Workload ラック内のすべての物理サーバーのステータスが表示されます。

図 *476 :* サーバーメンテナンス



サーバーまたはコンポーネントの交換に関連する手順

図 *477*: サーバーのメンテナンス手順

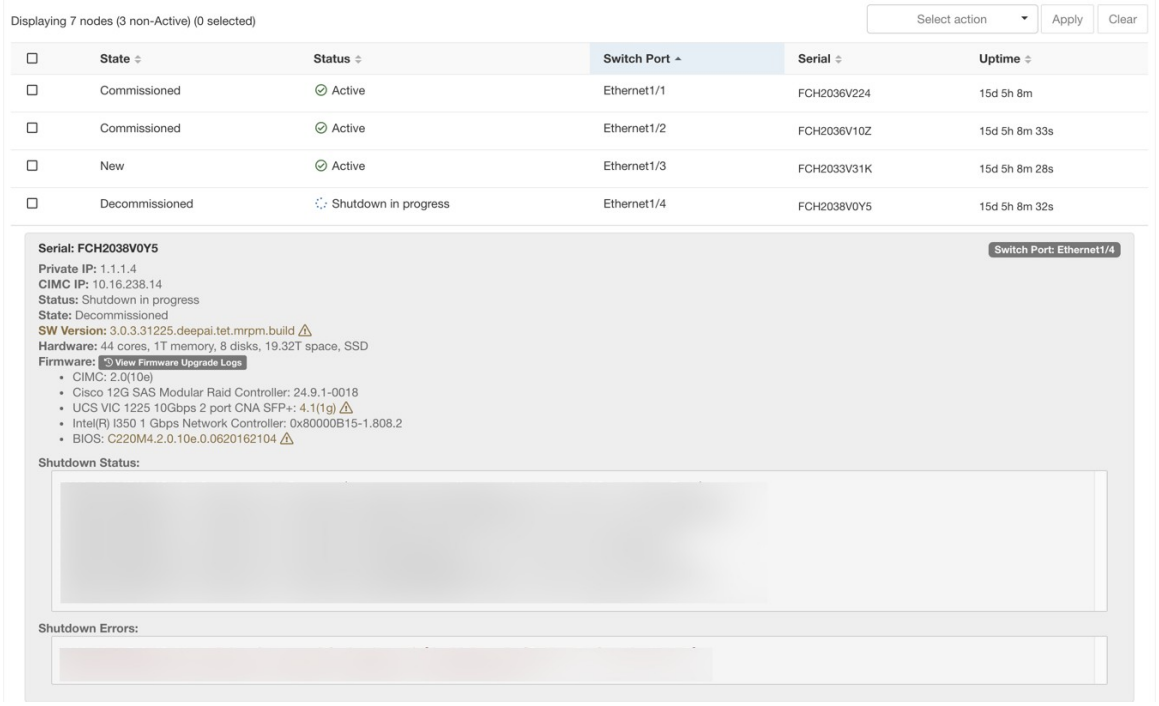


1. **メンテナンスが必要なサーバーの判断**：[クラスタステータス（Cluster Status）]ページで、サーバーの [シリアル（Serial）] 番号またはサーバーが接続されている [スイッチポート（Switchport）] を使用して判断します。交換するサーバーの CIMC IP を書き留めます。CIMC IP は、[クラスタステータス（Cluster Status ）] ページのサーバーボックスに表示されます。

2. **特別な VM のアクションの確認**：サーバーボックスから、サーバーに存在する VM またはインスタンスを見つけ、それらの VM に対して特別なアクションを実行する必要があるか確認します。次のセクションに、サーバーメンテナンス中の VM のアクションが一覧表示されています。

3. **サーバーのデコミッション**：デコミッション前のアクションが実行されたら、[クラスタステータス（Cluster Status）]ページを使用してサーバーをデコミッションします。サーバーに障害が発生し、ページに [非アクティブ（Inactive）] と表示された場合でも、サーバーのメンテナンス手順をすべて実行する必要があります。サーバーの電源がオフの場合でも、デコミッション手順を実行できます。

図 *478 :* サーバーのメンテナンス手順

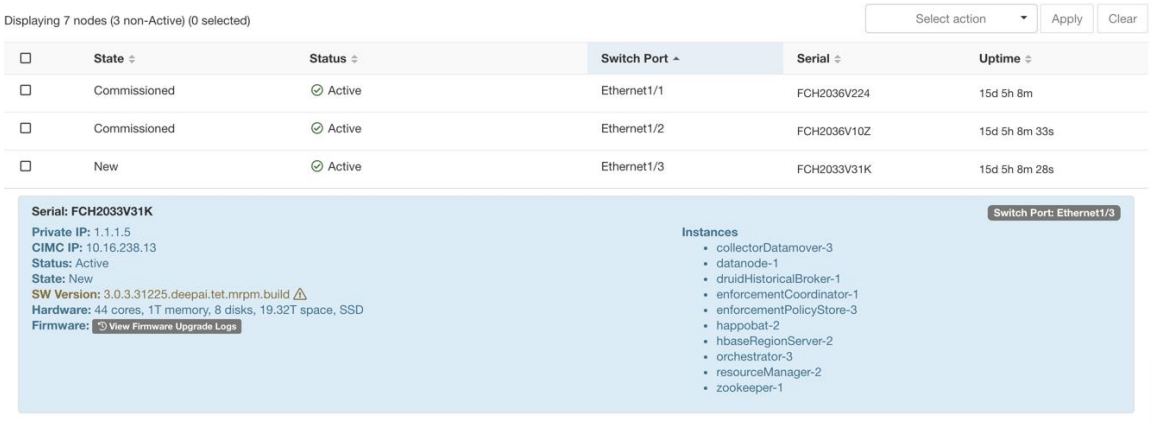


4. **サーバーのメンテナンスの実行**：[クラスタステータス（Cluster Status）] ページでノードが [デコミッション済み（Decommissioned）] とマークされたら、VM に対してデコミッション後の特別なアクションを実行します。これで、コンポーネントまたはサーバーを交換できます。サーバー全体を交換する場合は、新しいサーバーの CIMC IP を、交換したサーバーと同じ CIMC IP に変更します。各サーバーの CIMC IP は、[クラスタステータス（Cluster Status）] ページで確認できます。

5. **コンポーネント交換後の再イメージ化**：[クラスタステータス（Cluster Status）] ページを使用して、コンポーネント交換後にサーバーを再イメージ化します。再イメージ化には約 30 分かかり、サーバーへの CIMC アクセスが必要です。再イメージ化が完了したサーバーは [新規（NEW）] とマークされます。

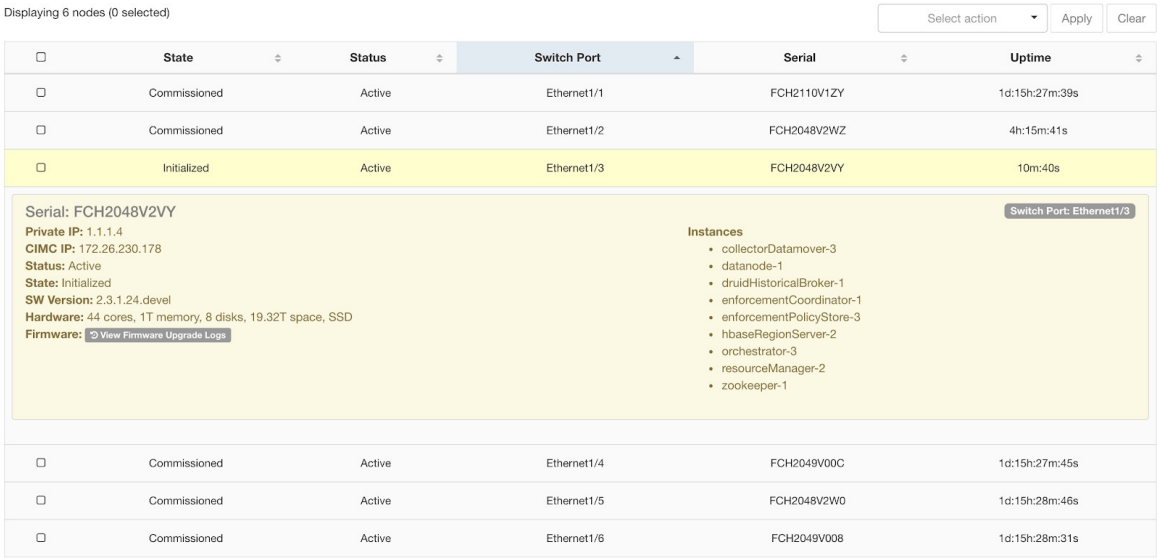
6. **サーバー全体の交換**：サーバー全体を交換した場合、そのサーバーは [クラスタステータス（Cluster Status）] ページに [新規（NEW）] 状態で表示されます。サーバーのソフトウェアバージョンは、同じページで確認できます。ソフトウェアバージョンがクラスタのソフトウェアバージョンと異なる場合は、サーバーを再イメージ化します。

図 *479* : サーバーのメンテナンス手順



7. **サーバーのコミッション**：サーバーが[新規（NEW）]とマークされたら、[クラスタステータス（Cluster Status）]ページからノードのコミッショニングを開始できます。この手順により、サーバー上にVMがプロビジョニングされます。サーバーのコミッショニングには約45分かかります。コミッショニングが完了すると、サーバーは[コミッション済み（Commissioned）]とマークされます。

図 *480* : サーバーのメンテナンス手順



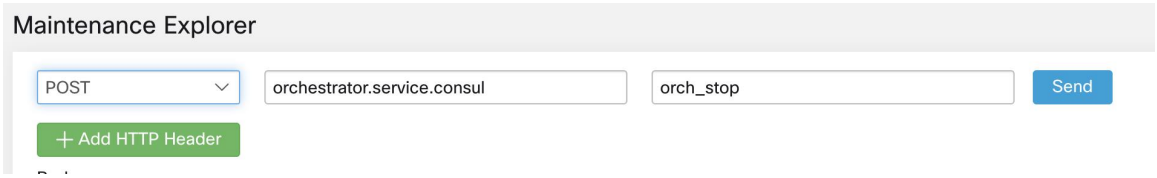
サーバーメンテナンス中の VM のアクション

一部のVMでは、サーバーのメンテナンス手順中に特別なアクションを実行する必要があります。それらのアクションは、デコミッション前、デコミッション後、またはコミッション後に実行できます。

1. **オーケストレータのプライマリ**：これはデコミッション前のアクションです。メンテナンス中のサーバーにプライマリオーケストレータがある場合は、デコミッションする前に探

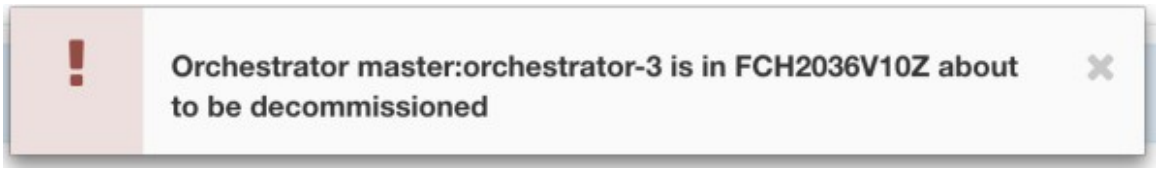
索ページから orchestrator.service.consul に orch_stop コマンドを POST します。これで、プライマリオーケストレータが切り替わります。

図 *481*：サーバーのメンテナンス手順



プライマリオーケストレータでサーバーをデコミッションしようとすると、次のエラーが表示されます。

図 *482*：サーバーのメンテナンス手順



オーケストレータのプライマリを特定するには、任意のホストで explore コマンド「primaryorchestrator」を実行します。

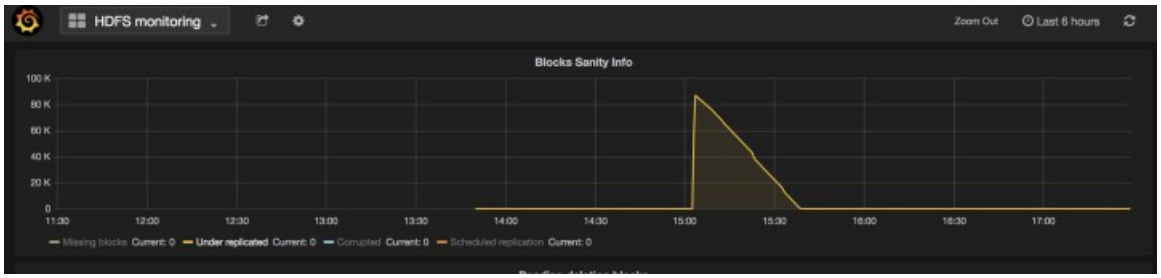
2. **NameNode**：メンテナンス中のサーバーに NameNode VM がある場合は、デコミッション後に探索ページから orchestrator.service.consul に switch_namenode を POST し、コミッション後に orchestrator.service.consul に switch_namenode を POST します。これらは、デコミッション後とコミッション後のアクションです。

3. **セカンダリ NameNode**：メンテナンス中のサーバーにセカンダリ NameNode VM がある場合、デコミッション後に探索ページから orchestrator.service.consul に switch_secondarynamenode を POST し、コミッション後に orchestrator.service.consul に switch_secondarynamenode を POST します。これらは、デコミッション後とコミッション後のアクションです。

4. **Resource Manager プライマリ**：メンテナンス中のサーバーに Resource Manager プライマリがある場合、探索ページから orchestrator.service.consul に switch_yarn を POST します。これらは、デコミッション後とコミッション後のアクションです。

5. **DataNode**：クラスタは、一度に 1 つの DataNode 障害のみを許容します。DataNode VM を持つ複数のサーバーにメンテナンスが必要な場合は、一度に 1 つずつサーバーのメンテナンスを実行します。各サーバーのメンテナンス後、[モニタリング（Monitoring）] | [hawkeye] | [hdfs-monitoring] | [健全性情報のブロック（Block Sanity Info）] の下に表示されるチャートの [欠落ブロック数（Missing blocks）] および [レプリケーション中（Under replicated）] の数が 0 になるのを待ちます。

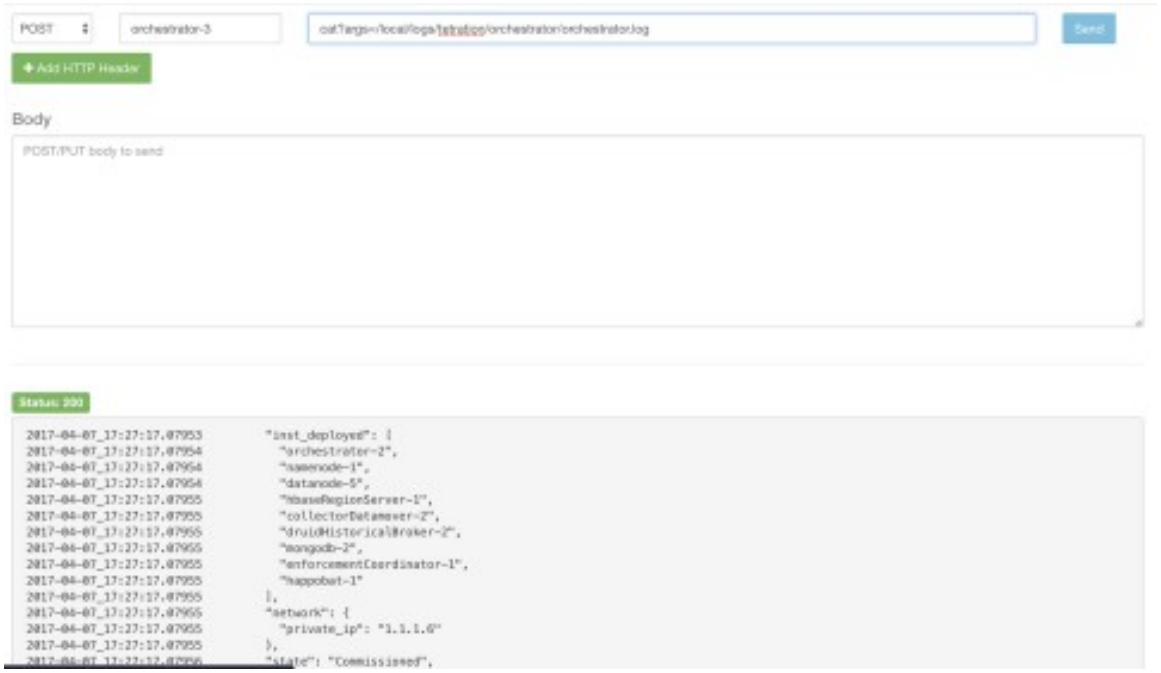
図 *483* : サーバーのメンテナンス手順



サーバーメンテナンスのトラブルシューティング

1. **ログ**：サーバーのメンテナンスログはすべて、オーケストレータログの一部で、orchestrator.service.consul の /local/logs/tetration/orchestrator/orchestrator.log にあります。

   図 *484* : サーバーのメンテナンスログ



2. **デコミッション**：
   1. この手順により、サーバー上の VM またはインスタンスが削除されます。
   2. 次に、バックエンド consul テーブル内にある削除されたインスタンスのエントリが削除されます。
   3. この手順は約 5 分かかります。
   4. 完了すると、サーバーは[デコミッション済み（Decommissioned）] とマークされます。

（注） デコミッションされても、サーバーの電源がオフになるわけではありません。デコミッションでは、サーバー上の Secure Workload コンテンツのみ削除されます。

5. 電源がオフになっているサーバーは[非アクティブ（Inactive）]とマークされます。このサーバーのデコミッションは、[クラスタステータス（Cluster Status）]ページから引き続き実行できます。ただし、サーバーの電源がオフになっているため、VM の削除手順は実行されないため、このサーバーがデコミッション状態でクラスタに再び参加しないようにしてください。サーバーは再イメージ化してクラスタに追加し直す必要があります。

3. **再イメージ化**：
   1. この手順では、サーバーに Secure Workload ベース OS またはハイパーバイザ OS をインストールします。
   2. また、ハードドライブをフォーマットし、サーバーにいくつかの Secure Workload ライブラリをインストールします。
   3. 再イメージ化では、**mjolnir** というスクリプトを実行して、サーバーのイメージングが開始されます。mjolnir の実行には約 5 分かかり、その後、実際のイメージングが開始されます。イメージングには約30分かかります。イメージング中のログは、再イメージ化されているサーバーのコンソールでのみ確認できます。ユーザーは、ta_dev キーを使用して、再イメージ化に関する追加情報を確認できます。PXE ブート時の /var/log/nginx ログ、DHCP IP および PXE ブート構成を確認するための /var/log/messages などがあります。
   4. 再イメージ化には、オーケストレータからの CIMC 接続が必要です。CIMC の接続を確認する最も簡単な方法は、探索ページを使用するか、orchestrator.service.consul から ping?args=<cimc ip> を POST する方法です。サーバーを交換した場合は、CIMC IP を変更し、CIMC パスワードをデフォルトのパスワードに設定することを**忘れないでください**。
   5. また、スイッチが正しいルートで設定されるように、クラスタの展開時に CIMC ネットワークがサイト情報に設定されている必要があります。クラスタ CIMC 接続が正しく設定されていない場合、オーケストレータログに次の結果が表示されます。

4. **コミッション**：
   1. コミッショニングでは、サーバー上の VM がスケジュールされ、VM でプレイブックを実行して Secure Workload ソフトウェアがインストールされます。
   2. コミッショニングが完了するまでに約 45 分かかります。
   3. ワークフローは、展開またはアップグレードのワークフローと同様です。
   4. ログには、コミッショニング中の障害が示されます。

5. [クラスタステータス（Cluster Status）] ページのサーバーは、コミッショニング中は [初期化済み（Initialized）] としてマークされ、手順完了後にのみ [コミッション済み（Commissioned）] としてマークされます。

ベアメタル除外

電源シャットダウン後のクラスタの再起動時にハードウェア障害が検出された場合、現在のところ、サービスを安定させるための再起動ワークフローの実行も、コミッションワークフローの実行もできない（サービスが停止するとコミッションが失敗するため）状態でクラスタがスタックします。この機能は、このようなシナリオで役立つことが期待されており、ユーザーは障害のあるハードウェアで再起動（アップグレード）でき、その後、失敗したベアメタルの通常の RMA プロセスを実行できます。

ユーザーは、POST を使用して、除外するベアメタルのシリアルでエンドポイントを探索する必要があります。

1. アクション：POST
2. ホスト：orchestrator.service.consul
3. エンドポイント：exclude_bms?method=POST
4. 本文：{“baremetal”：[“BMSERIAL”]}

オーケストレータは、除外が実行可能か判断するためにいくつかのチェックを実行します。この場合、オーケストレータはいくつかの consul キーをセットアップし、次の再起動またはアップグレードワークフローで除外されるベアメタルと VM を示す成功メッセージを返します。ベアメタルに特定の VM が含まれている場合、それらの VM は除外できません（以下の「制限事項」セクションを参照）。探索エンドポイントは、除外できない理由を示すメッセージで応答します。探索エンドポイントでの POST が成功すると、ユーザーはメイン UI から再起動またはアップグレードを開始し、通常どおり再起動を続行できます。アップグレードの最後に、除外 BM リストを削除します。BM を除外してアップグレードまたは再起動を再度実行する必要がある場合、ユーザーは bmexclude 探索エンドポイントに再度 POST する必要があります。

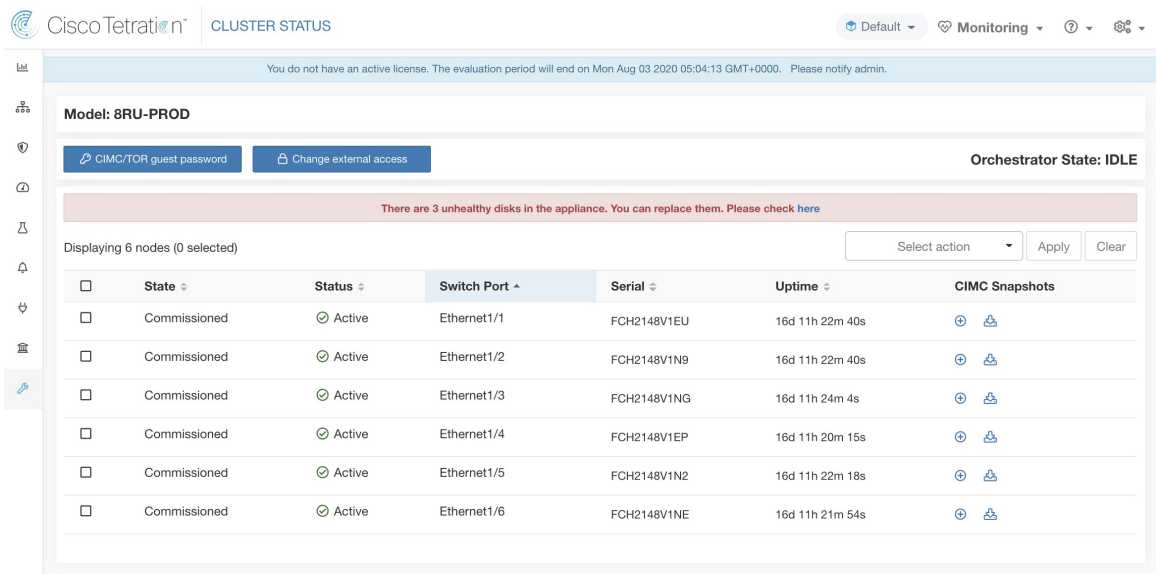
制限事項：現在、次の VM は除外できません。1. namenode 2. secondaryNamenode 3. mon- godb 4. mongodbArbiter

ディスク メンテナンス

ディスクメンテナンスには、サーバーから障害のあるハードディスクを交換することが含まれます。オーケストレータは、クラスタ内のすべてのサーバーで bmmgr によって報告されるディスクの状態を監視します。障害のあるディスクが検出された場合、[クラスタステータス（Cluster Status）] ページ（左側のナビゲーションバーの [トラブルシューティング（Troubleshoot）] メニューからアクセスできます）のバナーにそれが示されます。このバナーには、異常（UNHEALTHY）状態のディスクの数が表示されます。そのバナーでこちらをクリックすると、ディスク交換ウィザードが表示され、ディスクメンテナンスのすべての手順を実行するこ

とができます。[クラスタステータス（Cluster Status）] ページと同様に、ディスク交換ページにはすべてのユーザーがアクセスできますが、アクションを実行できるのはカスタマーサポート ユーザーのみです。

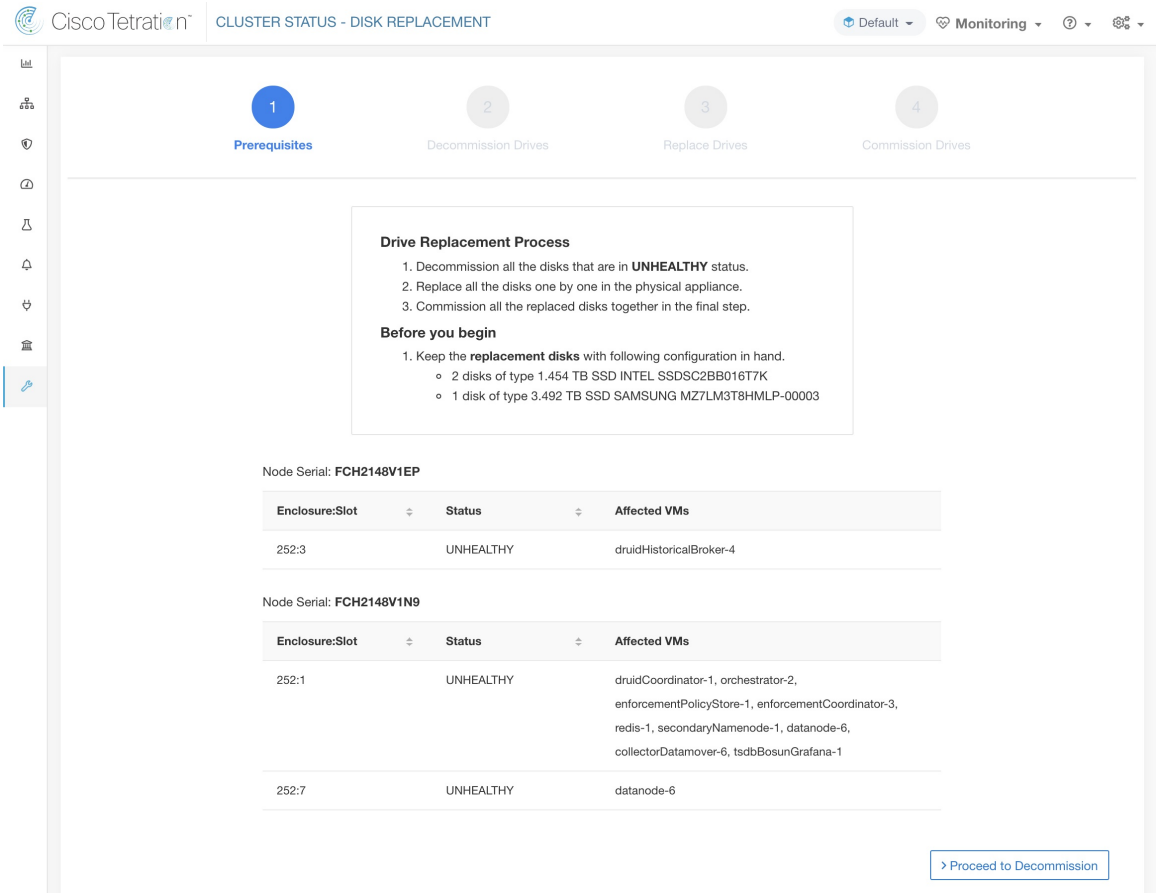
図 *485 :* 障害のあるディスクのバナー



ディスク交換ウィザード

ディスク交換ウィザードのランディングページには、障害が発生したディスクの詳細が表示されます。これらの詳細には、交換が必要なすべてのディスクのサイズ、タイプ、製造元、およびモデルが含まれます。また、スロット ID も表示され、これらの各ディスクを使用するすべての VM が一覧表示されます。交換プロセスを開始する前に、交換用ディスクを用意しておく必要があります。

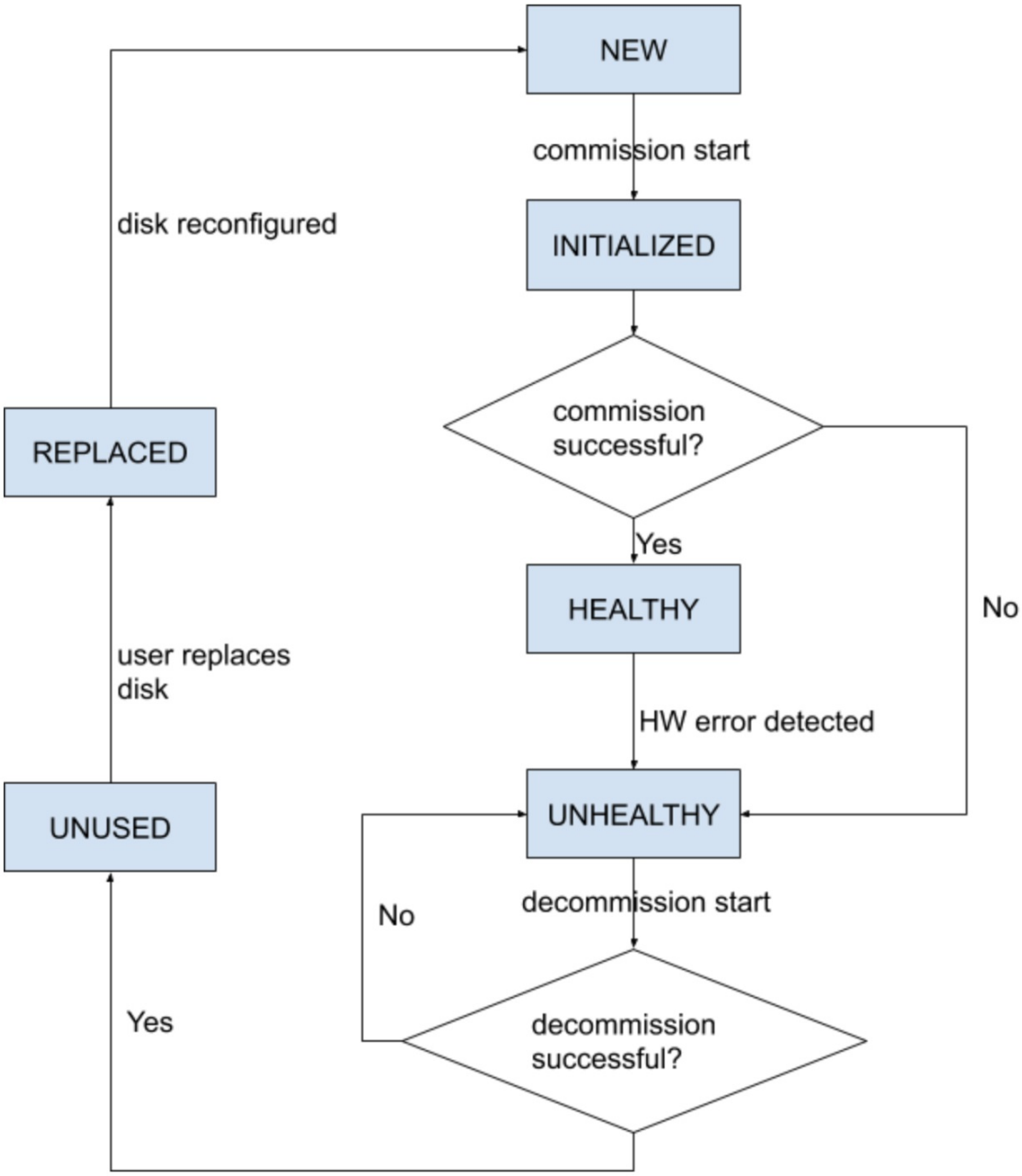
図 *486* : ディスク交換ウィザード



ディスクのステータス遷移

クラスタでは、ハードディスクには次の 6 つの状態があります。正常（HEALTHY）、異常（UNHEALTHY）、未使用（UNUSED）、交換済み（REPLACED）、新規（NEW）、初期化済み（INITIALIZED）。展開/アップグレード後のクラスタ内のすべてのディスクのステータスは正常（HEALTHY）です。さまざまなエラーが検出され、1 つ以上のディスクのステータスが異常（UNHEALTHY）になることがあります。

図 *487* : ディスクのステータス遷移



ディスク交換プロセスの最初のステップは、デコミッションです。この手順では、これらのディスクを使用するすべての仮想マシンをクラスタから削除します。デコミッションされたディスクのステータスは未使用（UNUSED）になります。デコミッション後、交換用ディスクを適切なスロットに挿入する必要があります。ディスクが交換されたことをユーザーが確認します。これは、新しくインストールされたディスクを再設定するためのバックエンド信号になります。するとステータスが交換済み（REPLACED）に変わり、次のハードウェアスキャン後

に、これらの交換されたディスクのステータスが新規（NEW）に変わります。この移行には 2 ～ 3 分かかることがあります。

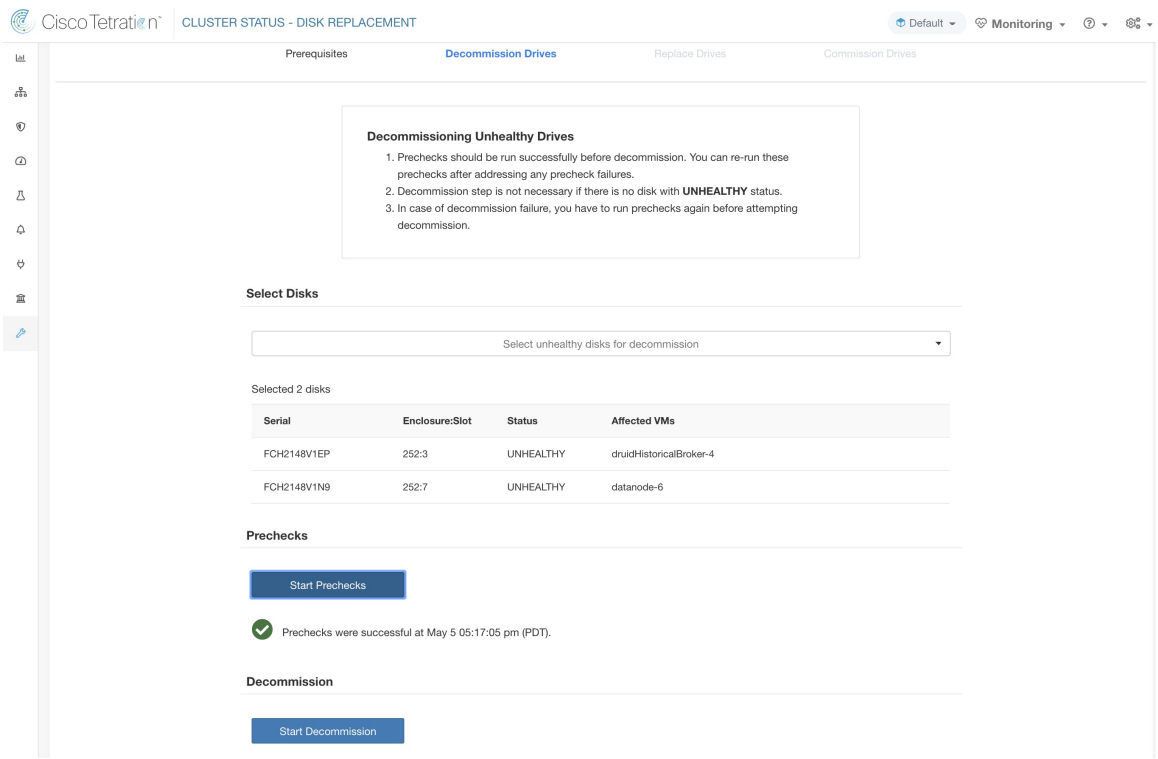
すべてのディスクを交換して再設定したら、コミッショニングに進み、デコミッションプロセスの一部として削除されたすべての VM を展開します。コミッションが開始されると、ディスクのステータスが初期化済み（INITIALIZED）に変わります。コミッションが成功すると、すべてのディスクのステータスが正常（HEALTHY）になります。このステップで失敗すると、ステータスは再び異常（UNHEALTHY）になり、デコミッションからの回復を再度開始することになります。

要件の事前チェック

デコミッションまたはコミッションの手順を実行する前に、要件の事前チェックを実行する必要があります。バックエンドでさまざまなチェックが実行されますが、ユーザーがデコミッションまたはコミッションのステップに進む前に、これらのチェックすべてに合格する必要があります。失敗したチェックは、失敗の詳細と推奨される修正処置とともにディスク交換ウィザードで報告されます。必要な手順を続行する前にこれらの修正処置を実行する必要があります。

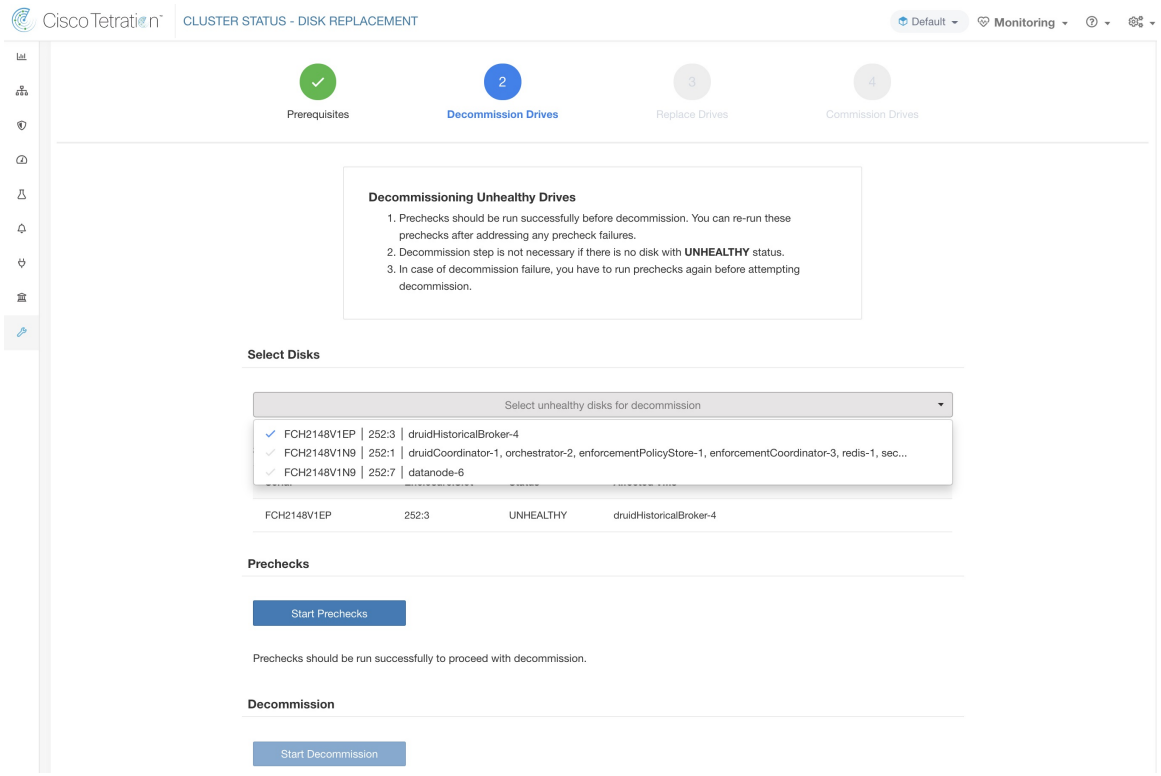
こうした事前チェックの例は次のとおりです。namenode と secondaryNamenode を一緒にデコミッションすることはできません。一度にデコミッションできるデータノードは 1 つだけです。 namenode はコミッション前に正常でなければなりません。

図 *488* : ディスク交換の事前チェック



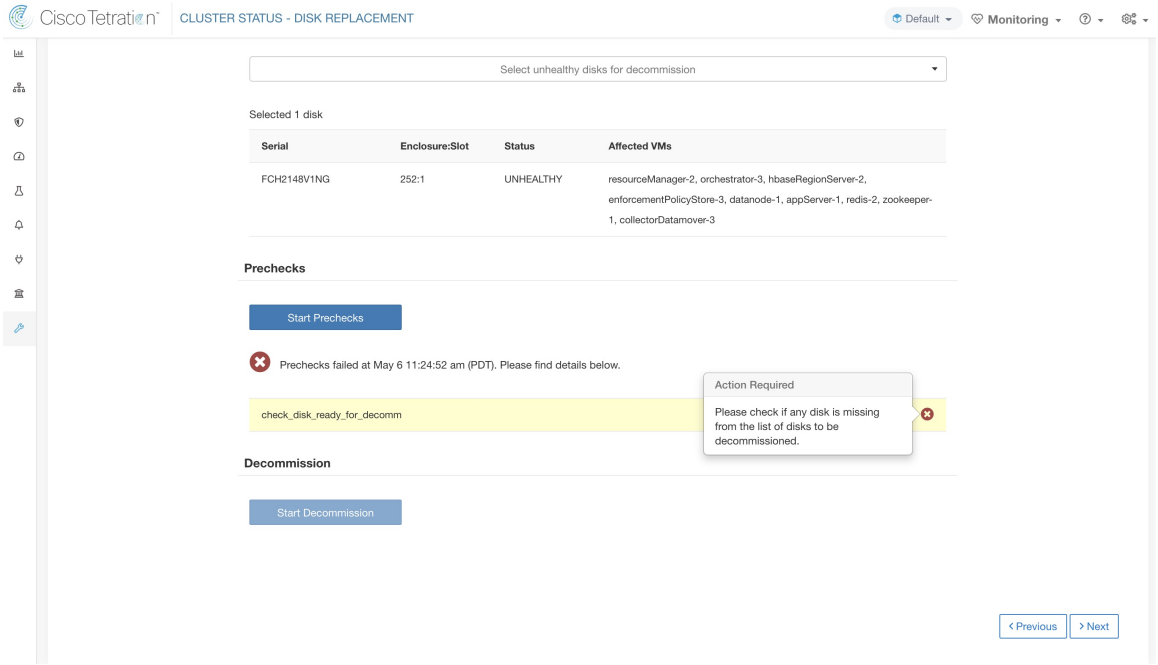
ユーザーは、障害が発生したディスクの任意のセットをまとめてデコミッションするために選択し、デコミッションの事前チェックを開始できます。障害が発生したディスクのセットを変更するには、事前チェックを再実行する必要があります。タスク（デコミッション/コミッション）を開始する前に、同じ事前チェックを再度行って、最後の事前チェックの実行からデコミッションタスクの開始までの間に新しい事前チェックの失敗がないことを確認します。

図 *489* : デコミッションする *1* つまたはすべての異常（*UNHEALTHY*）ディスクの選択



事前チェックが失敗した場合、失敗メッセージをクリックすると詳細メッセージが表示され、ポインターを赤い十字ボタンの上に置くと、推奨される処置がポップオーバーに表示されます。

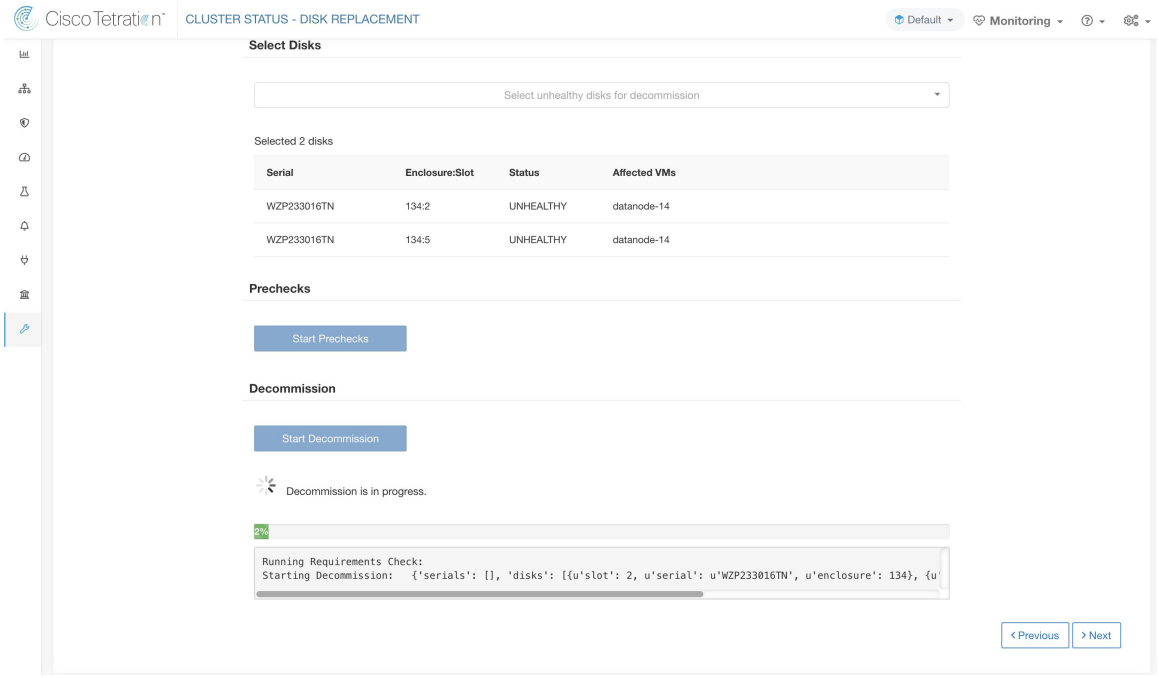
図 *490* : 事前チェックに失敗した場合にポップオーバーで表示される推奨処置



デスクのデコミッション

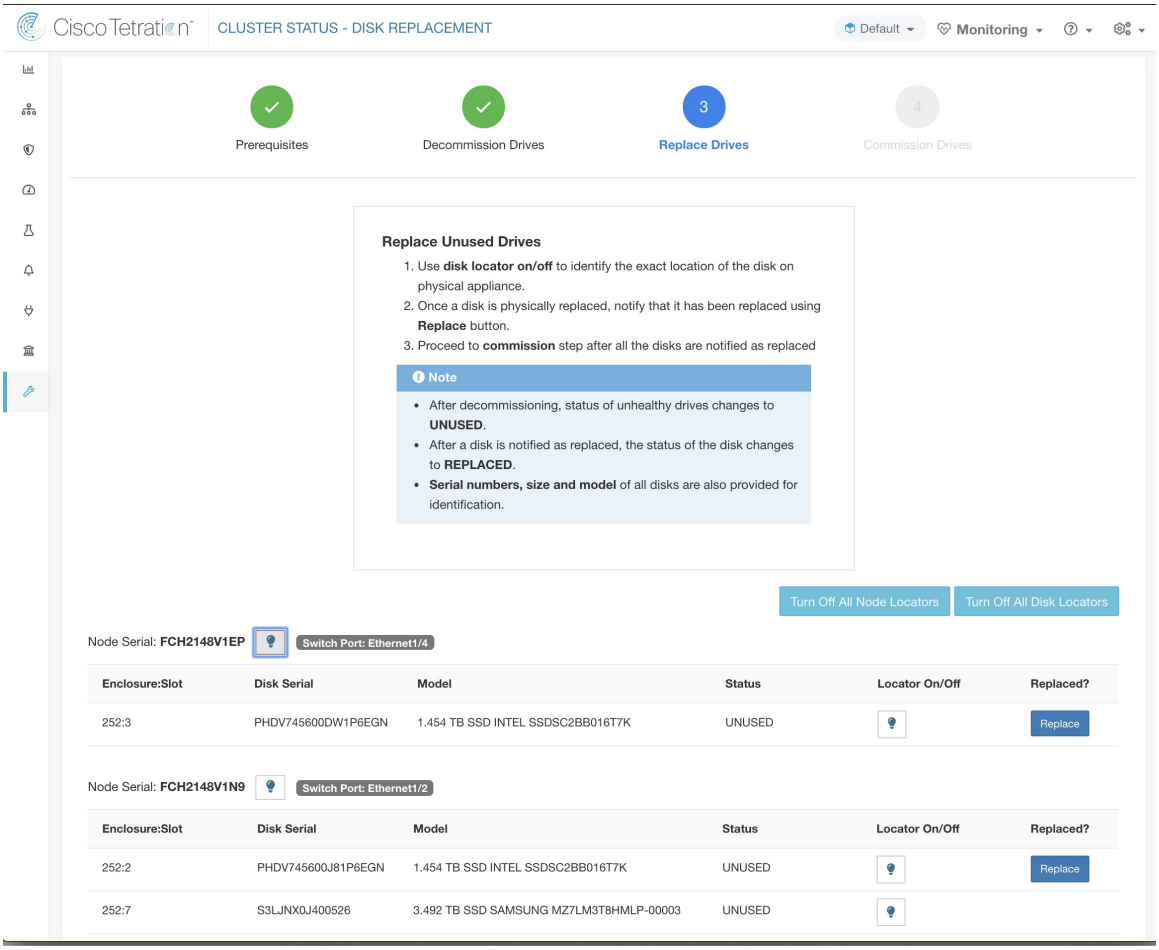
事前チェックに合格すると、ユーザーはディスクのデコミッションに進むことができます。デコミッションの進行状況は、ディスク交換ウィザードに表示されます。デコミッションの進行状況が 100% に達すると、デコミッションされたすべてのディスクのステータスが未使用（UNUSED）に変わります。

図 *491* : ディスクのデコミッションの進行状況の監視



ディスクの交換

図 *492*：新しく追加されたディスクの再設定



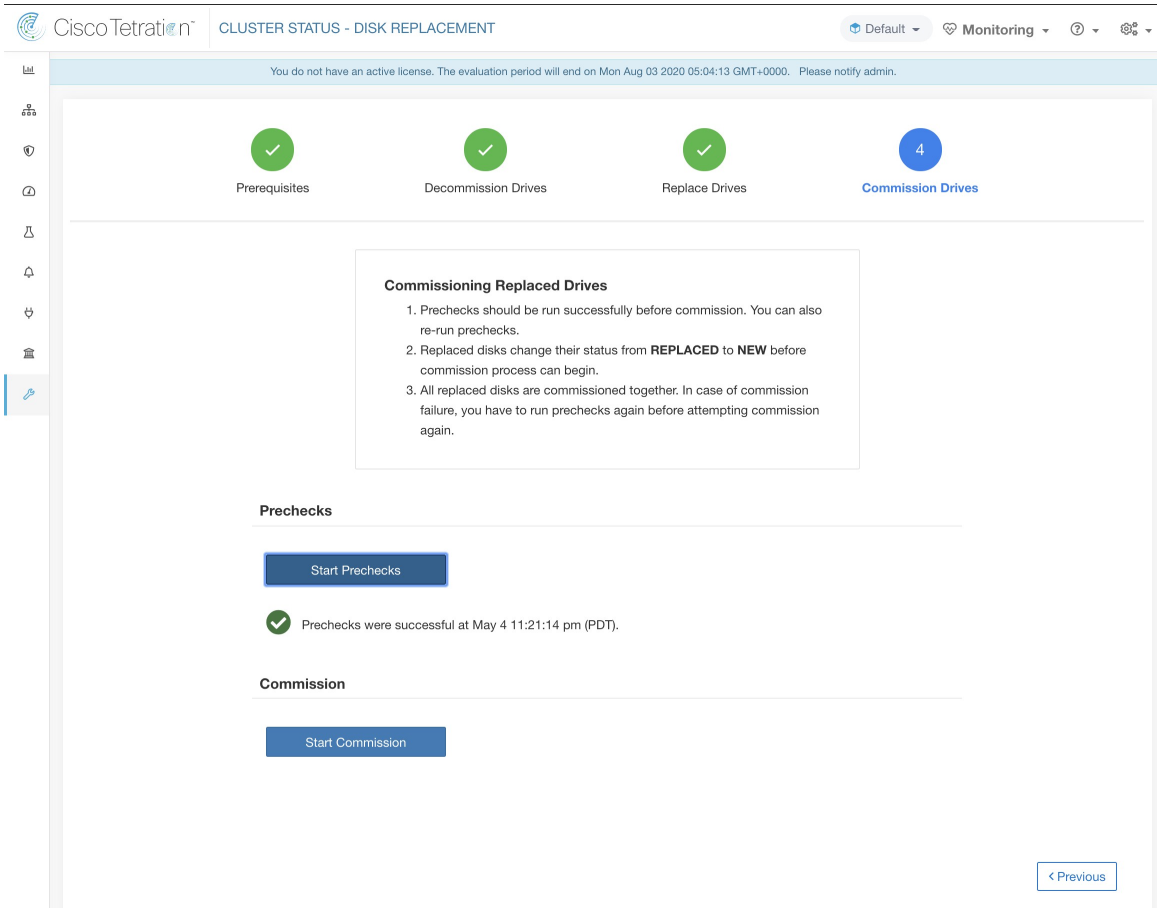
ディスクのデコミッション後、ユーザーはディスクを物理的に交換する必要があります。このプロセスを支援するために、交換ページにディスクとサーバーのロケーター LED へのアクセス機能を追加しました。サーバーとディスクのロケーター LED をすべてオフにするボタンがあるので、他のプロセスでロケーターがオンのままになっている可能性に対処することができます。

ディスクの物理的な交換は任意の順序で行えますが、再設定は特定のサーバーの最小スロット番号から最大スロット番号の順序で行う必要があります。この順序は、UI とバックエンドの両方に適用されます。UI では、ステータスが未使用（UNUSED）で、スロット番号が最も小さいディスクの交換ボタンがアクティブになります。

デスクのコミッショニング

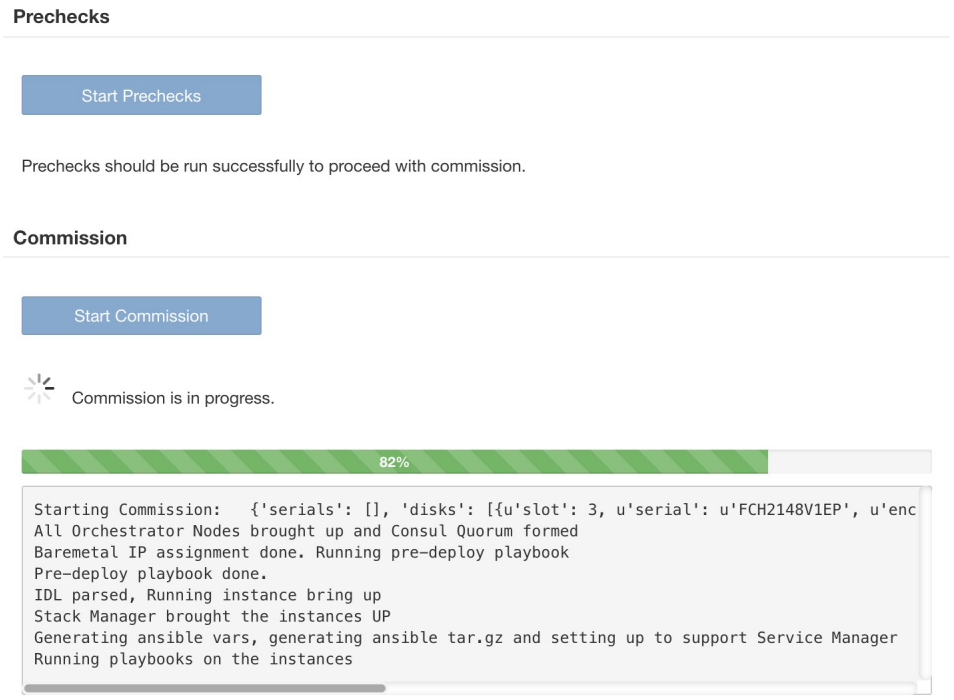
すべてのディスクを交換したら、コミッションに進みます。デコミッションと同様に、コミッション転を続行する前に一連の事前チェックを実行する必要があります。

図 *493* : コミッション前の事前チェック



コミッションの進行状況は、ディスクコミッションページで監視します。コミッションが正常に終了すると、すべてのディスクのステータスが正常（HEALTHY）に変わります。

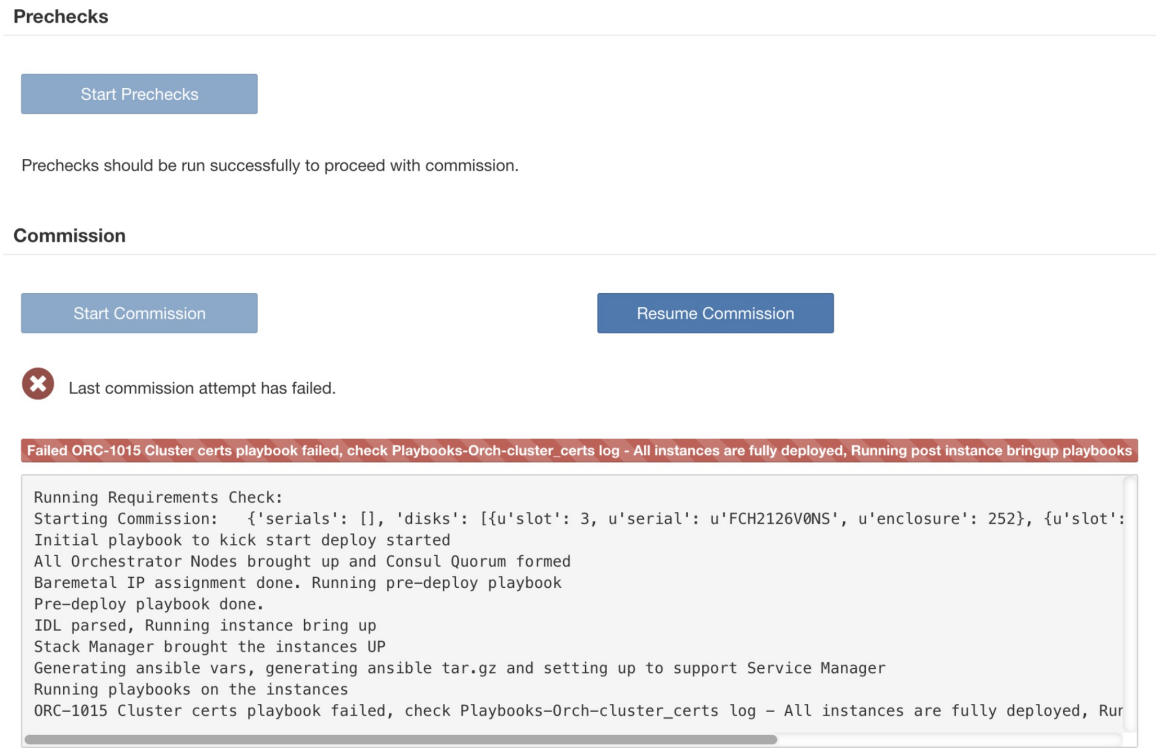
図 *494* : コミッションの進行状況



コミッション中の障害からの復旧

VM が再展開された後に障害が発生した場合は、再開（Resume）機能で回復できます。このようなエラーが発生した場合、[コミッションの再開（Resume Commission）] ボタンがディスクコミッションページに表示されます。このボタンをクリックすると、展開後のプレイブックを再起動してコミッションを続行できます。

図 *495 :* コミッションの再開



VM が再展開される前に障害が発生した場合、コミッション中だったディスクのステータスは異常（UNHEALTHY）に変更されます。そのため、交換プロセスは異常（UNHEALTHY）ディスクのデコミッションから再開する必要があります。

コミッション中の追加のディスク障害

ディスクのコミッションの進行中に交換対象のディスク以外のディスクに障害が発生した場合、進行中のコミッションプロセスが成功または失敗した後に、ディスク交換ウィザードにこの障害に関する通知が表示されます。

再開可能な障害が発生した場合、ユーザーは次のステップについて2つのオプションから選択することができます。

1. 現在のコミッションを再開および完了してから、新しい障害に対するディスク交換プロセスの実行を試みます。
2. または、新しく故障したディスクのデコミッションを開始し、すべてのディスクのコミッションをまとめて実行します。

この2番目の方法は、再開不可能な障害が発生した場合に使用できる唯一の方法です。新しく障害が発生したディスクが原因で展開後に障害が発生した場合、再開ボタンは使用できますが、その場合でも 2 番目の方法が唯一の対処方法になります。

トラブルシューティング

ログ

1. すべてのディスクコミッション/デコミッションログは、オーケストレータログの一部です。デバッグの開始ポイントは、orchestrator.service.consul の /local/logs/tetration/orchestrator/orchestrator.log である必要があります。
2. ディスクの交換/再設定アクション中に発生した障害の詳細は、対象のサーバーの bmmgr ログで探すことができます。サーバー上のログの場所は、/local/logs/tetration/bmmgr/bmmgr.log になります。

制限事項

1. サーバーのルートボリュームを含むディスクは、この手順では交換できません。このようなディスク障害は、サーバーメンテナンスプロセスを使用して修正する必要があります。
2. ディスクのコミッションは、すべてのサーバーがアクティブで、コミッション済みの状態にある場合にのみ実行できます。ディスクとサーバーの交換を組み合わせることが必要な場合の対応方法については、以下の特別な対処方法のセクションを参照してください。

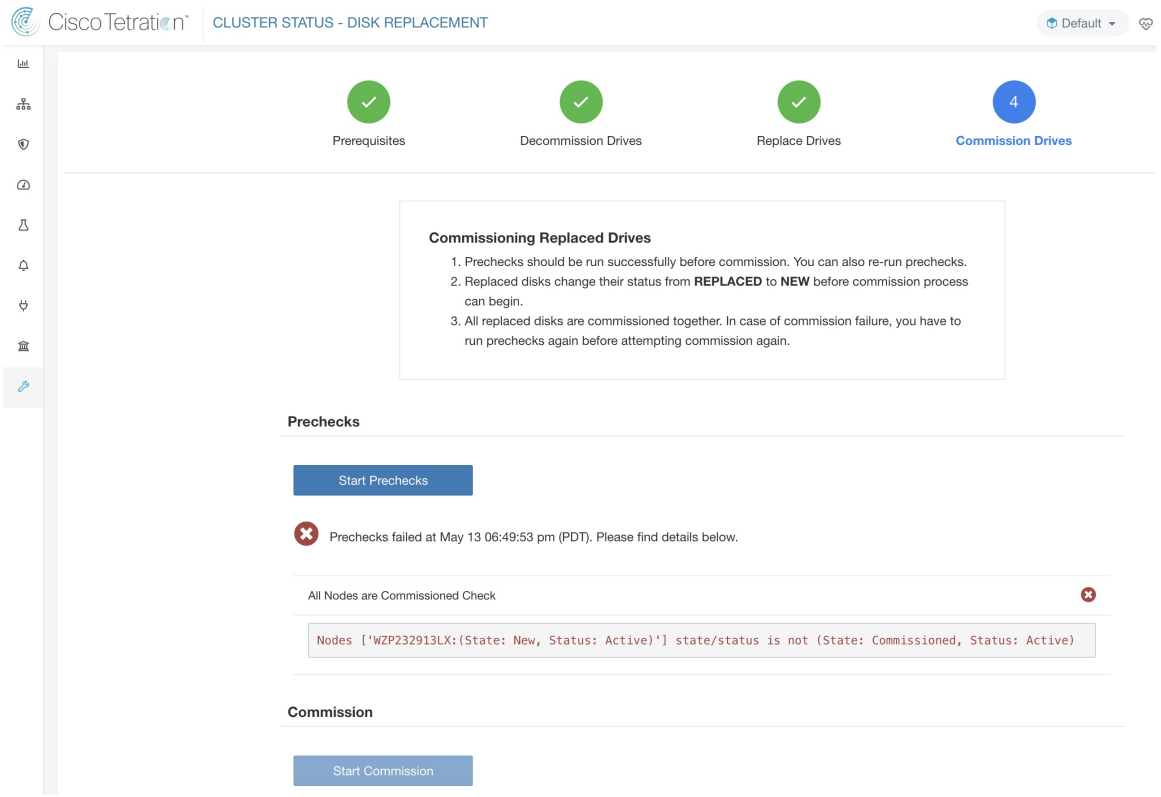
特別な対処方法

ディスクとサーバーをまとめて交換する

ディスクとサーバーを同時にコミッションする必要がある障害シナリオでは、ユーザーは、デコミッション可能なすべてのディスクをデコミッションして交換する必要があります。事前チェックで以下ことが確認されるため、これらのディスクをコミッションすることはできません。

1. すべての正常でないディスクのステータスが新規（NEW）であること。
2. すべてのサーバーのステータスがアクティブで、コミッション済みの状態であること。

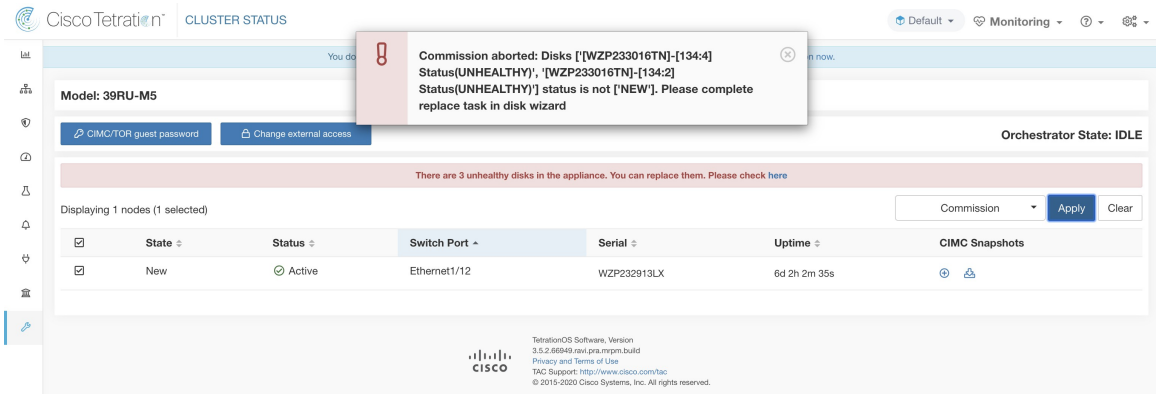
図 *496*：ディスクのコミッションの前に、すべてのサーバーがコミッション済みでアクティブであることを確認します



すべての異常（UNHEALTHY）ディスクが新規（NEW）状態になると、障害の発生したサーバーは、サーバーメンテナンス手順を使用してデコミッション/再イメージ化/再コミッションさせることが期待されます。

これで、ステータスが正常（HEALTHY）または新規（NEW）でないディスクがある場合、サーバーのコミッションが防止されます。サーバーのコミッションが成功すると、すべてのディスクのステータスも正常（HEALTHY）になります。

図 *497*：サーバーをコミッションする前に、障害のあるすべてのディスクが新規（***NEW***）状態であることを確認します



クラスタのメンテナンス：クラスタのシャットダウンと再起動

このセクションでは、クラスタ全体に影響を及ぼす2つのメンテナンス操作について説明します。

1. クラスターのシャットダウン
2. クラスターの再起動

クラスタのシャットダウン

クラスタのシャットダウンでは、実行中のすべてのSecure Workloadプロセスを停止させ、個々のノードの電源をすべてダウンさせます。以下の手順でシャットダウンを実行してください。

シャットダウンの開始

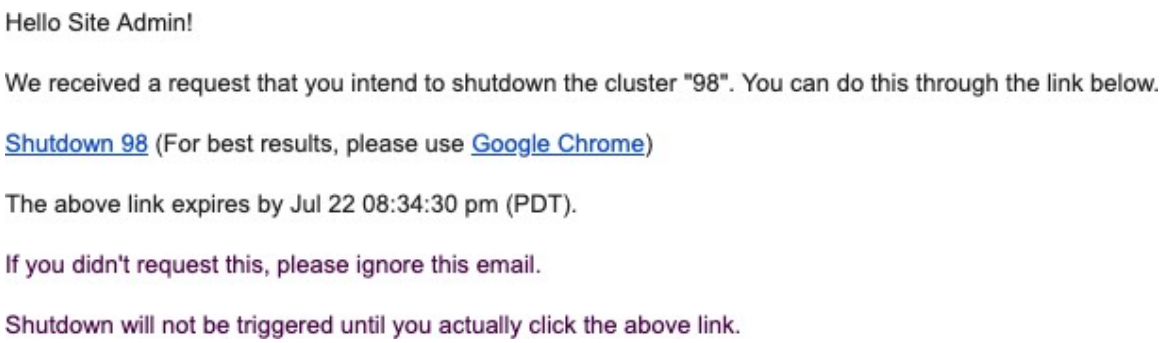
手順

ステップ1 ウィンドウの左側にあるナビゲーションバーから、**[プラットフォーム（Platform）]** > **[アップグレード/再起動/シャットダウン（Upgrade/Reboot/Shutdown）]**をクリックします。

ステップ2 [再起動/シャットダウン（Reboot/Shutdown）] タブをクリックします。

ステップ3 [シャットダウン（Shutdown）] オプションボタンを選択し、[シャットダウンリンクの送信（Send Shutdown Link）] をクリックします。クリックすると、以下に示すように、電子メールでシャットダウンリンクが送信されます。シャットダウンリンクは、リンクを要求しているユーザーの電子メールアドレスに配信されます。

図 *498 :* シャットダウンメール


Hello Site Admin!

We received a request that you intend to shutdown the cluster "98". You can do this through the link below.

Shutdown 98 (For best results, please use Google Chrome)

The above link expires by Jul 22 08:34:30 pm (PDT).

If you didn't request this, please ignore this email.

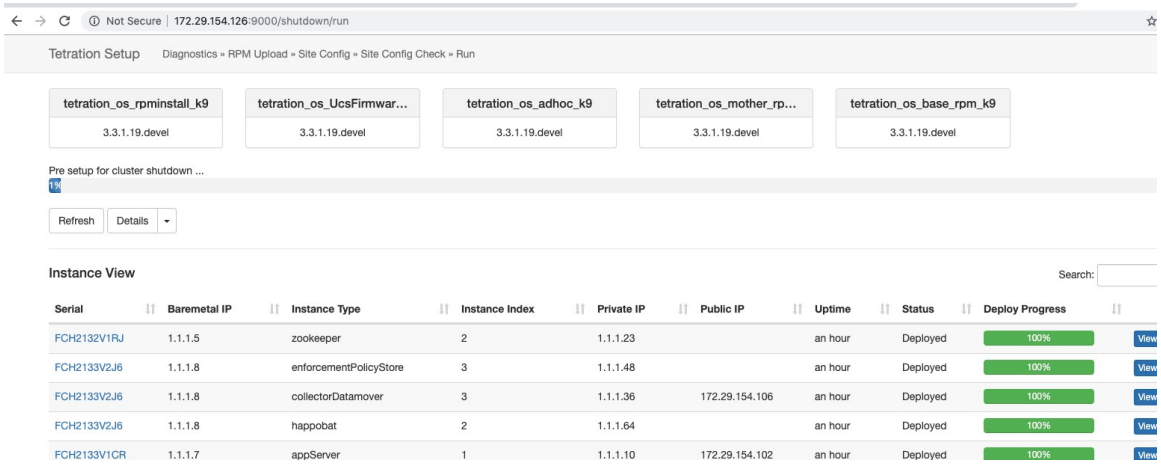
Shutdown will not be triggered until you actually click the above link.

ステップ 4 [クラスタシャットダウン（Cluster Shutdown）] ページの赤い [シャットダウン（Shutdown）] ボタンをクリックして、シャットダウンを開始します。**重要：このボタンをクリックした後にシャットダウンをキャンセルすることはできません。**

シャットダウンの進捗状況

シャットダウンが開始されると、シャットダウンの進行状況を追跡する進行状況バーがページに表示されます。

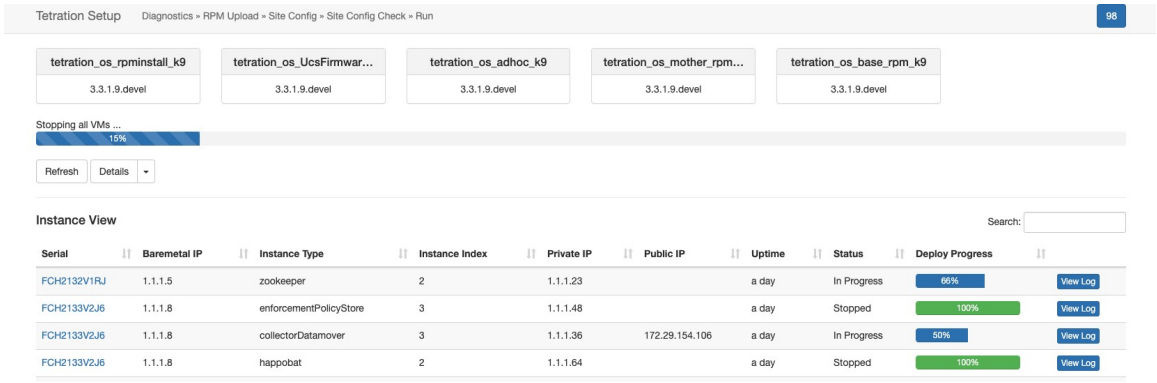
図 *499 :* シャットダウンの進捗状況



最初のシャットダウンの事前チェックでエラーが発生した場合、進行状況バーが赤くなり、エラーの修正後にクリックしてシャットダウンを再開できる再開ボタンが表示されます。

事前チェックが完了すると、VMが停止します。VMが徐々に停止する間、その進行状況がページの下部に表示されます。このページは、アップグレード中の VM 停止のページに似ています。表示されている各フィールドの詳細については、アップグレードのセクションを参照してください。VM の停止には最大 30 分かかる場合があることに注意してください。

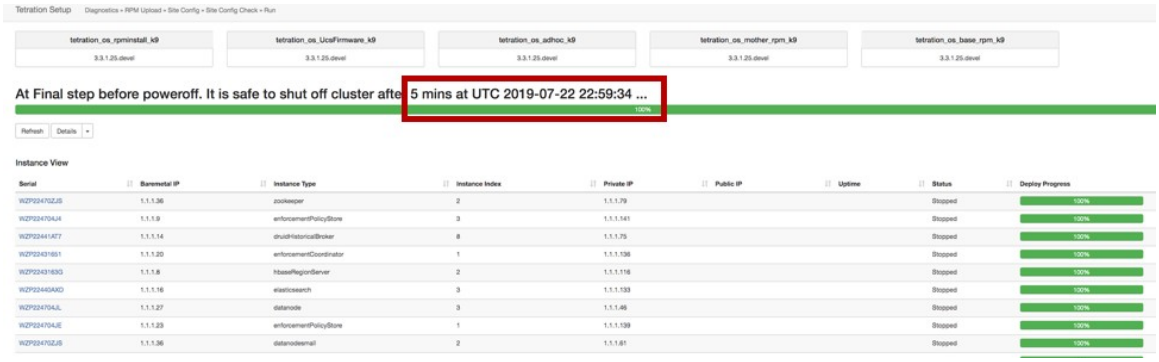
図 *500 : VM* 停止



最終的に、クラスタを完全にシャットダウンする準備が整うと、進行状況バーが 100% になり、クラスタの電源の安全な切断が可能になる時刻が示されます。その時刻は、以下のスクリーンショットで強調表示されています。

（注） 進行状況バーに表示される時刻を過ぎるまで、クラスタの電源を切らないでください。

図 *501 :* シャットダウン *100%*



クラスタの再起動

シャットダウン後にクラスタを回復するには、ベアメタルの電源をオンにします。個々のベアメタルがすべて起動すると、UI に再びアクセスできるようになります。クラスタにログインした後、クラスタを再起動して、クラスタを再び完全に動作可能な状態にする必要があります。

（注） クラスタを再び完全に動作可能な状態にするには、シャットダウン後にクラスタを再起動する必要があります。

再起動の開始

手順

ステップ 1 ウィンドウの左側にあるナビゲーションバーから、**[プラットフォーム（Platform）]** > **[アップグレード/再起動/シャットダウン（Upgrade/Reboot/Shutdown）]**をクリックします。

ステップ 2 [再起動/シャットダウン（Reboot/Shutdown）] タブをクリックします。

ステップ 3 [再起動（Reboot）] ラジオボタンを選択し、[再起動リンクを送信（Send Reboot Link）] をクリックします。

再起動リンクは、リンクを要求しているユーザーの電子メールアドレスに送信されます。

Secure Workload サービスの再起動は、制限付きのアップグレード操作を実行します。電子メールの再起動リンクをクリックすると、再起動を開始できるセットアップ UI に移動します。

ここから先の手順はアップグレードと同じです。詳細については、アップグレードセクションを参照してください。

シャットダウンと再起動の履歴

シャットダウンと再起動の履歴は、[アップグレード（Upgrade）] ページの [履歴（History）] タブに表示されます（左側のナビゲーションバーから**[プラットフォーム（Platform）]**>**[アップグレード/再起動/シャットダウン（Upgrade/Reboot/Shutdown）]**からアクセスします）。

[データタップ管理者（Data Tap Admin）]：データのタップ

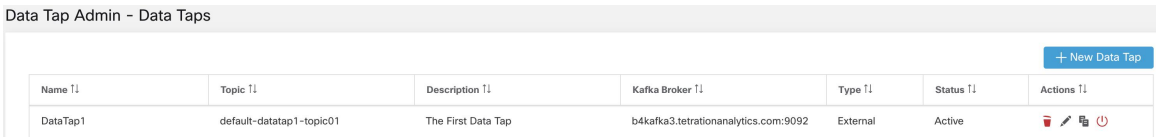
1. データタップ
2. 管理対象データタップ

データタップ

（注） Cisco Secure Workload は現在、データタップ用に Kafka Broker 0.9.x、0.10.x、1.0.x、1.1.x への書き込みをサポートしています。

Secure Workload クラスタからアラートをプッシュするには、ユーザーは設定済みのデータタップを使用する必要があります。データタップ管理ユーザーは、新規または既存のデータタップを設定およびアクティブ化できる唯一のユーザーです。ユーザーは、自分の[テナント（Tenant）]に属するデータタップのみを表示できます。

図 *502*：利用可能なデータタップ



データタップを管理するには、ウィンドウの左側にあるナビゲーションバーで **[管理（Manage）]** > **[データタップ管理者（Data Tap Admin）]** をクリックします。

推奨される Kafka 設定

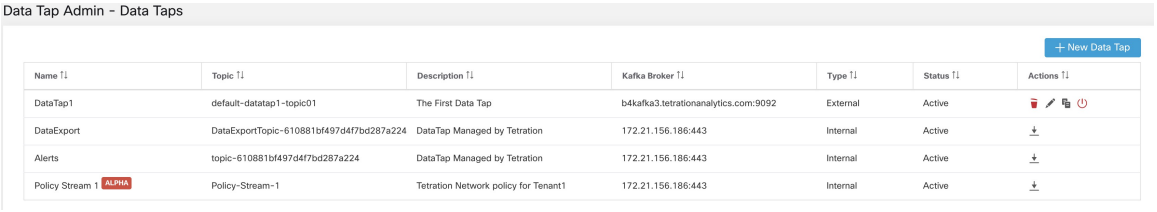
Kafka クラスタを設定する際は、Secure Workload では 9092、9093 または 9094 以降のポートの使用が推奨されます。これらのポートは Secure Workload が Kafka の発信トラフィック用に開くポートであるためです。Kafka Broker の推奨設定は次のとおりです。

```
  broker.id=<incremental number based on the size of the cluster>
  auto.create.topics.enable=true
  delete.topic.enable=true
  listeners=PLAINTEXT://:9092
  port=9092
  default.replication.factor=2
  host.name=<your_host_name>
  advertised.host.name=<your_adversited_hostname>
  num.network.threads=12
  num.io.threads=12
  socket.send.buffer.bytes=102400
  socket.receive.buffer.bytes=102400
  socket.request.max.bytes=104857600
  log.dirs=<directory where logs can be written, ensure that there is sufficient space
to hold the kafka journal logs>
  num.partitions=72
  num.recovery.threads.per.data.dir=1
  log.retention.hours=24
  log.segment.bytes=1073741824
  log.retention.check.interval.ms=300000
  log.cleaner.enable=false
  zookeeper.connect=<address of zookeeper ensemble>
  zookeeper.connection.timeout.ms=18000
```

データタップ管理セクション

[データタップ管理者（Data Tap Admins）] は、**[管理（Manage）]** > **[データタップ管理者（Data Tap Admin）]** > **[データタップ（Data Taps）]** ページに移動して、利用可能なすべてのデータタップを表示および設定できます。データタップは [テナント（Tenant）] ごとに設定されます。

図 *503 :* 利用可能なすべてのデータタップ

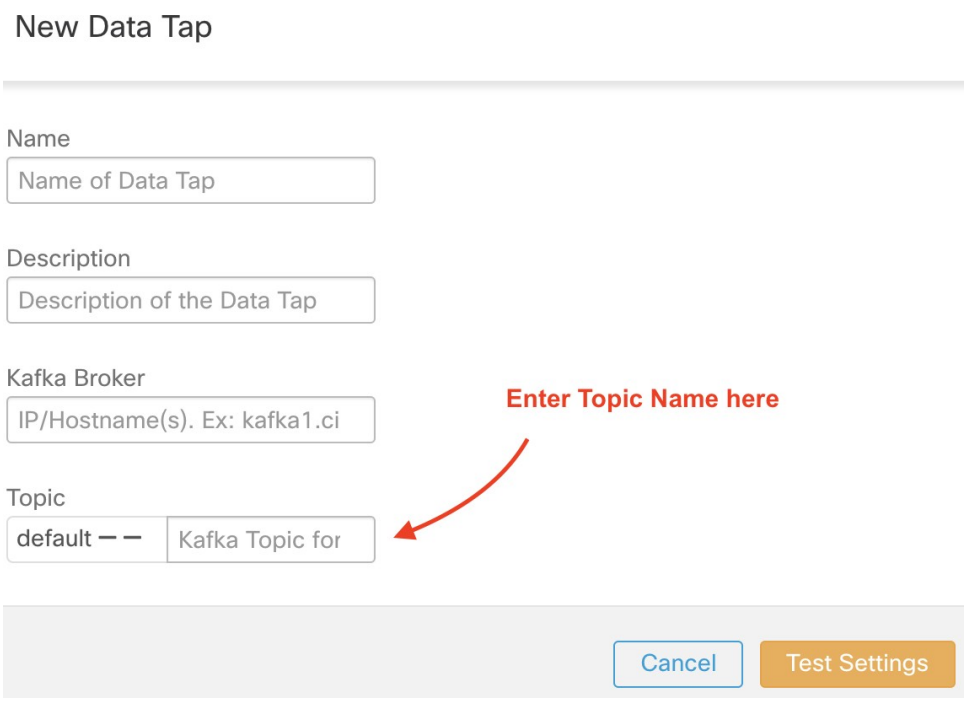


新しいデータタップの追加

データタップ管理者は、 

をクリックして新しいデータタップを追加できます

図 *504*：新しいデータタップの追加

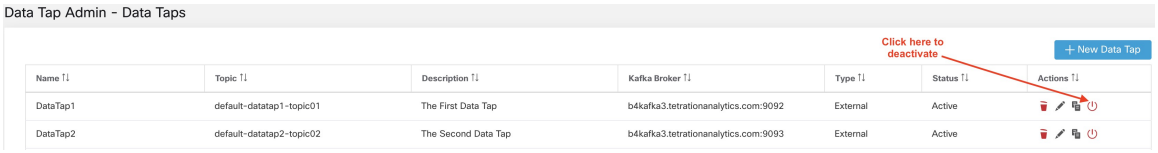


（注） データタップの値を変更するには、設定を検証する必要があります。

データタップの非アクティブ化

データ管理者は、一時的にSecure Workloadからメッセージが送信されないように、データタップを非アクティブ化できます。そのデータタップへのメッセージは送信されません。データタップはいつでも再開できます。

図 *505*：データタップの非アクティブ化



データタップの削除

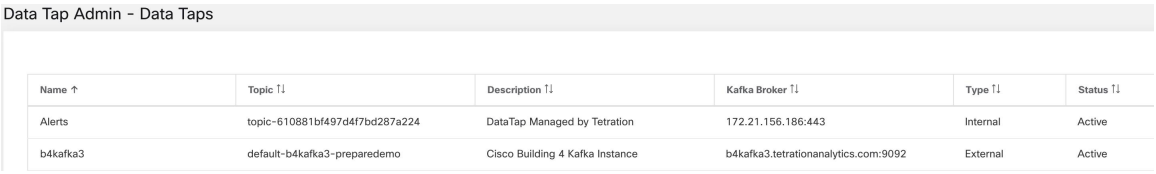
データタップを削除すると、そのアプリケーションに依存するすべてのSecure Workloadアプリケーションインスタンスが削除されます。たとえば、ユーザーがコンプライアンスアラートを（Secure Workloadアプリケーションアラートで）データタップAに送信するように指定し、管理者がデータタップAを削除した場合、アラートアプリケーションはデータタップAをアラート出力対象にしなくなります。

管理対象データタップ

管理対象データタップ（MDT）は、Secure Workload クラスタ内でホストされるデータタップです。認証、暗号化、承認に関しては十分に安全です。MDT との間でメッセージを送受信するには、クライアントを認証する必要があり、ネットワーク経由で送信されるデータは暗号化され、承認されたユーザーのみが Secure Workload MDT との間でメッセージを読み書きできます。 Secure Workload は、UI からダウンロードされるクライアント証明書を提供します。 Secure Workload は Apache Kafka 1.1.0 をメッセージブローカとして使用し、クライアントに同じバージョンと互換性のある安全なクライアントの使用を推奨します。

MDT はルート範囲の作成時に自動的に作成されます。すべてのルート範囲には、作成されたアラート MDT があります。ユーザーは Secure Workload クラスタからアラートを引き出すためにアラート MDT を使用する必要があります。証明書をダウンロードできるのはデータタップ管理ユーザーのみです。ユーザーは、[ルート範囲（root scope）] に属する MDT のみを表示できます。

図 *506*：設定されたデータタップのリスト



すべての Secure Workload アプリケーションアラートはデフォルトで MDT に送信されますが、別のデータタップに変更できます。証明書をダウンロードするには、2 つの選択肢があります。

1. JKS（Jave キーストア形式）。JKS 形式は Java クライアント向きです。
2. Certs。通常の証明書は、Go クライアントで簡単に使用できます。

図 *507*：ダウンロード（*Download*）

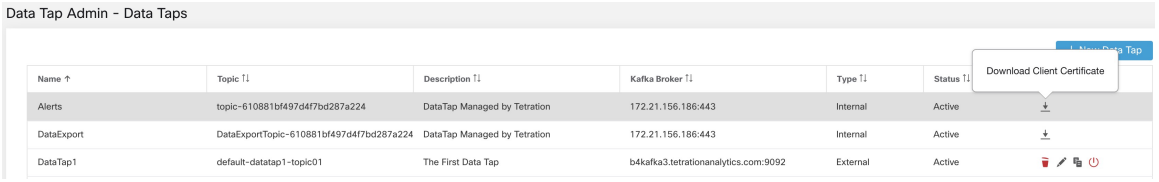
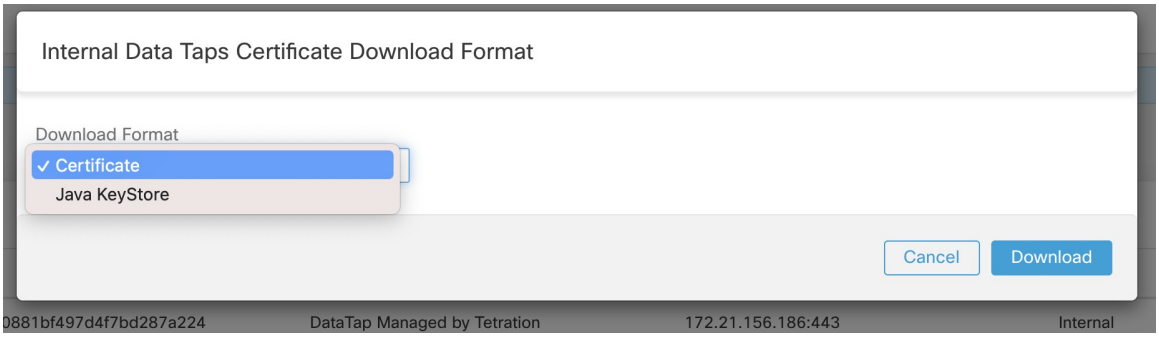


図 *508*：証明書の種類



Java キーストア

Alerts.jks.tar.gz をダウンロードすると、Secure Workload MDT に接続してメッセージを受信するための情報を含む次のファイルが表示されます。

1. kafkaBrokerIps.txt：このファイルには、Kafka クライアントが Secure Workload MDT への接続に使用する必要がある IP アドレス文字列が含まれています。
2. topic.txt：このファイルには、このクライアントによるメッセージの読み取りが可能なトピックが含まれています。トピックは <root_scope_id> のトピック形式です。この root_scope_id は、後で Java クライアントで他のプロパティを設定するときに使用できます。
3. keystore.jks：Kafka クライアントが以下に示す接続設定で使用するキーストアです。
4. truststore.jks：Kafka クライアントが以下に示す接続設定で使用するトラストストアです。
5. passphrase.txt：このファイルには、#3 と #4 で使用するパスワードが含まれています。

キーストアとトラストストアを使用する Consumer.properties（Java クライアント）を設定する際には、次の Kafka 設定を使用する必要があります。

```
security.protocol=SSL
ssl.truststore.location=<location_of_truststore_downloaded>
ssl.truststore.password=<passphrase_mentioned_in_passphrase.txt>
ssl.keystore.location=<location_of_truststore_downloaded>
ssl.keystore.password=<passphrase_mentioned_in_passphrase.txt>
ssl.key.password=<passphrase_mentioned_in_passphrase.txt>
```

Java コードで Kafka コンシューマを設定する際には、次の一連のプロパティを使用する必要があります。

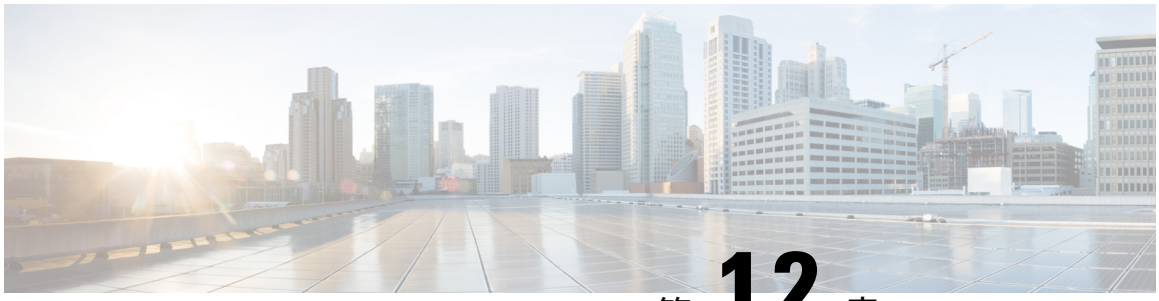
```
    Properties props = new Properties();
    props.put("bootstrap.servers", brokerList);
    props.put("group.id", ConsumerGroup-<root_scope_id>);  // root_scope_id is same as
mentioned above
    props.put("key.deserializer",
"org.apache.kafka.common.serialization.StringDeserializer");
    props.put("value.deserializer",
"org.apache.kafka.common.serialization.StringDeserializer");
    props.put("enable.auto.commit", "true");
    props.put("auto.commit.interval.ms", "1000");
    props.put("session.timeout.ms", "30000");
    props.put("security.protocol", "SSL");
    props.put("ssl.truststore.location", "<filepath_to_truststore.jks>");
    props.put("ssl.truststore.password", passphrase);
    props.put("ssl.keystore.location", <filepath_to_keystore.jks>);
    props.put("ssl.keystore.password", passphrase);
    props.put("ssl.key.password", passphrase);
    props.put("zookeeper.session.timeout.ms", "500");
    props.put("zookeeper.sync.time.ms", "250");
    props.put("auto.offset.reset", "earliest");
```

証明書

エンドユーザーが証明書を使用する場合は、Sarama Kafka ライブラリを使用している Go クライアントを使用して Secure Workload MDT に接続できます。Alerts.cert.tar.gz をダウンロードすると、次のファイルが表示されます。

1. kafkaBrokerIps.txt：このファイルには、Kafka クライアントが Secure Workload MDT への接続に使用する必要がある IP アドレス文字列が含まれています。
2. topic：このファイルには、このクライアントによるメッセージの読み取りが可能なトピックが含まれています。トピックは<root_scope_id>のトピック形式です。この root_scope_id は、後で Java クライアントで他のプロパティを設定するときに使用できます。
3. KafkaConsumerCA.cert：このファイルには、Kafka コンシューマの証明書が含まれています。
4. KafkaConsumerPrivateKey.key：このファイルには、Kafka コンシューマの秘密鍵が含まれています。
5. KafkaCA.cert：このファイルは、Go クライアントの root CA 証明書リストで使用する必要があります。

Secure Workload MDT に接続する Go クライアントの次の例を参照してください。（サンプル Go コードを添付）MDT からのアラートを使用するサンプル Go クライアント



第 12 章

モニタリング

使用できる**モニタリング**オプションは、ユーザーロールによって異なります。

- エージェントのモニタリング （919 ページ）
- 適用ステータス （923 ページ）
- ポリシー更新の一時停止 （925 ページ）
- ライセンス （925 ページ）

エージェントのモニタリング

このページには、現在選択されているルート範囲に基づいて、クラスタ内のすべての監視対象エージェントの数が表示されます。

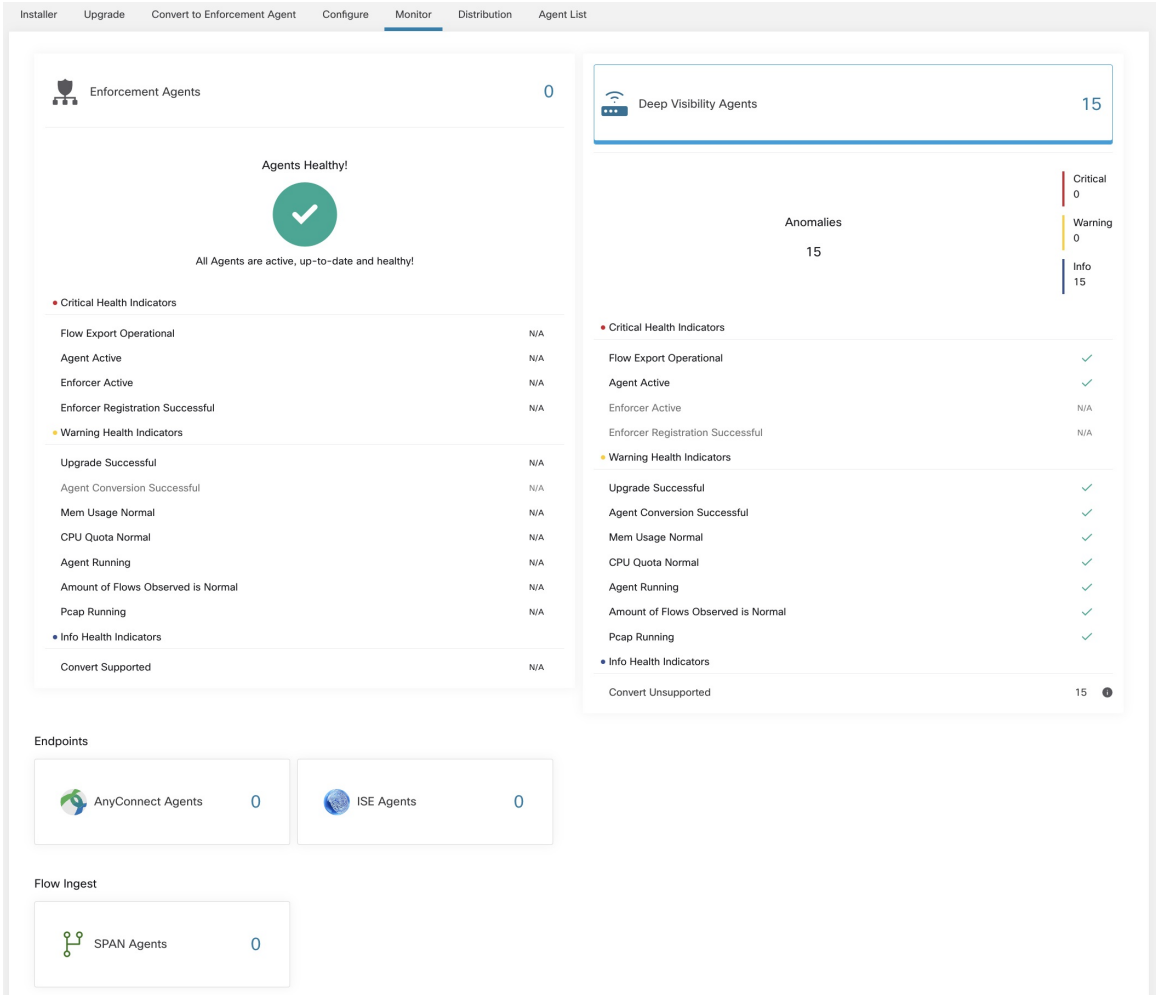
（注） インベントリの総数は、収集ルールを適用した後にネットワーク上で観察されたすべてのインベントリの合計です。

エージェントのモニタリング

エージェントを監視するには、左側のナビゲーションバーで**[管理（Manage）]** > **[エージェント（Agents）]**をクリックし、[監視（Monitor）] タブをクリックします。

このページは、**サイト管理者**および**カスタマーサポート**の役割を持つユーザーのみが利用できます。**範囲所有者**は、インベントリ、優れた可視性エージェント、および適用エージェントを表示できます。

図 *509* : インストールされているエージェントの総数



次の表は、エージェントタイプごとの違いを示しています。

Agent Type	説明
優れた可視性	時系列フローデータ、ホストで実行されるプロセスに関して最高の忠実度を提供します。ほとんどの Linux および Windows プラットフォームがサポートされています。sw_agents_deployment-label を参照してください。
施行	優れた可視性エージェントで使用可能なすべての機能を提供します。それに加えて、適用エージェントはインストールされているホストに対してファイアウォールルールを設定することができます。

AnyConnect	Network Visibility Module（NVM）を備えた AnyConnect セキュア モビリティ エージェントを実行しているエンドポイントで時系列フローデータを提供します。Cisco Secure Workload エージェントのインストールは必要ありません。NVM によって生成された IPFIX レコードは、Secure Workload AnyConnect プロキシコネクタに送信されます。Windows、Mac、および特定のスマートフォンのプラットフォームがサポートされています。
ISE	Cisco ISE に登録されているエンドポイントに関するメタデータを提供します。ISE コネクタは、ISE pxGrid を介してメタデータを収集し、ISE エージェントが ISE アプライアンスから取得した属性とエンドポイントにログインしたユーザーの LDAP 属性に基づいてラベルをプッシュするときに ISE エンドポイントを Secure Workload に登録します。
次の表は、Cisco Secure Workload が提供するさまざまなアプライアンスエージェントの概要を示しています。	
アプライアンスエージェント	説明
SPAN	ホストごとのエージェントのインストールを必要とせずに、フロー分析を提供します。Secure Workload ERSPAN VM アプライアンスで実行されます。任意の Cisco スイッチから発信された ERSPAN パケットを消費します。

（注） NetFlow、NetScaler、F5、AWS、AnyConnect Proxy などのアプライアンスエージェントが、コネクタとしてサポートされるようになりました。コネクタの詳細については、「コネクタとは」を参照してください。

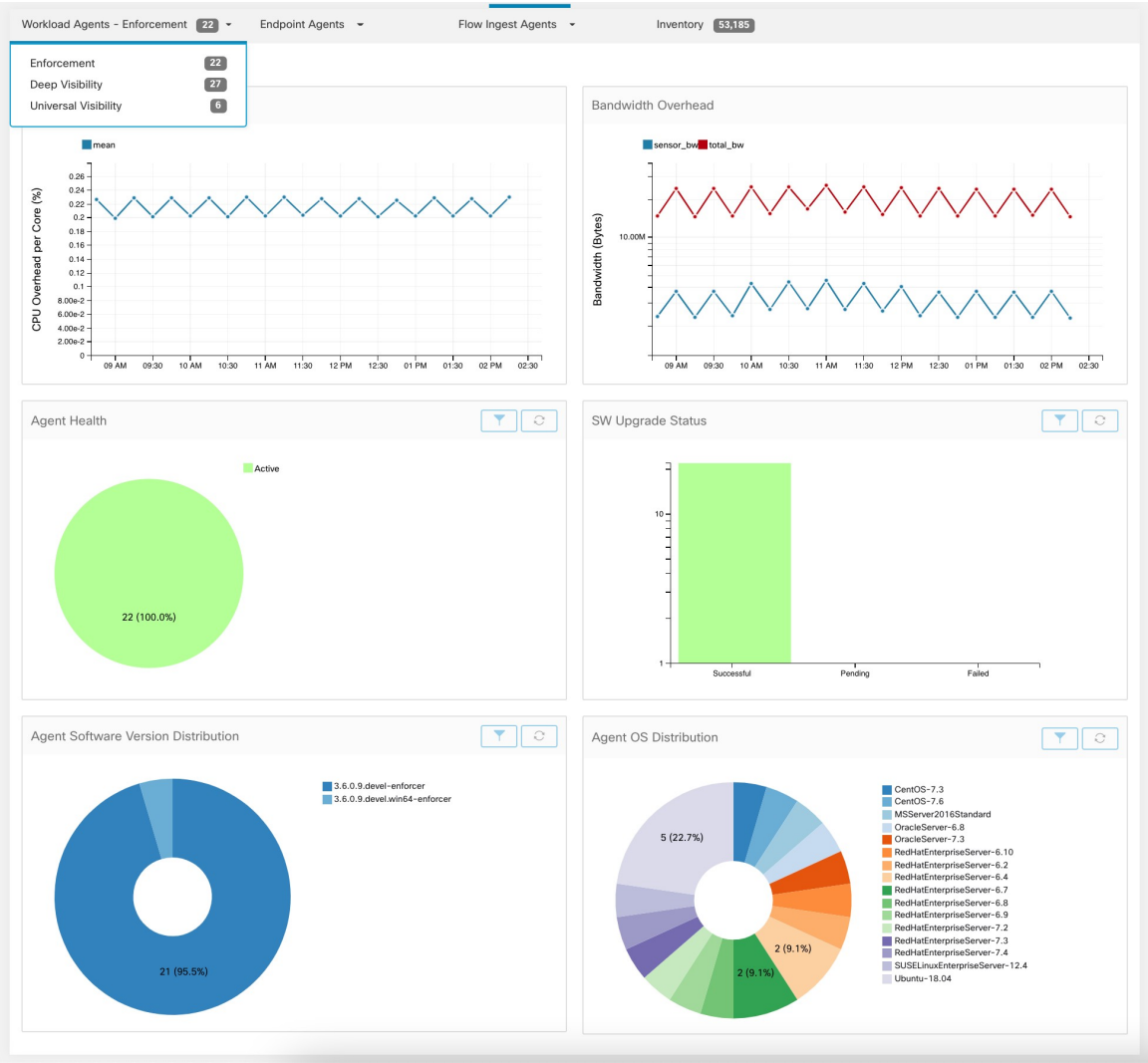
ゼロ以外のエージェントタイプのボタンを押すと、各エージェントタイプの分布にさらにドリルダウンできます。

エージェントのステータスと統計

このトピックで説明されているチャートを表示するには、[管理（Manage）] > [エージェント（Agents）] を選択し、[分布（Distribution）] タブをクリックします。

次のすべてのグラフは、詳細可視性タイプと適用エージェントタイプの両方で使用できます。

図 *510* : エージェントの分布



このページには、エージェントタイプごとに、全体的な CPU オーバーヘッド、帯域幅のオーバーヘッド、欠落したパケット、OS/バージョンの分布、エージェントのアップグレードステータスなど、登録されたエージェントの概要と正常性が表示されます。

[CPUオーバーヘッド（CPU Overhead）] チャート

[CPUオーバーヘッド（CPU Overhead）] チャートには、全エージェントからのコアごとのCPUオーバーヘッド集計ビューが表示されます。エージェントごとのCPUオーバーヘッドは、ワークロードプロファイルの一部として表示されます。このチャートは、詳細可視性タイプと適用エージェントタイプでのみ使用できます。

[帯域幅オーバーヘッド（Bandwidth Overhead）] チャート

[帯域幅オーバーヘッド（Bandwidth Overhead）] グラフには、総帯域幅とエージェントが使用する帯域幅の集約された統計が表示されます。エージェントごとの帯域幅オーバーヘッドは、

ワークロードプロファイルの一部として表示されます。このチャートは、詳細可視性タイプと適用エージェントタイプでのみ使用できます。

[エージェントの正常性（Agent Health）] チャート

[エージェントの正常性（Agent Health）] チャートには、アクティブ/非アクティブなエージェントの数が表示されます。アクティブなエージェントは、アップグレードのためにコンフィギュレーションサーバーに定期的にチェックインするエージェントです。チェックインの間隔は 30 分です。エージェントが 2 回を超えてチェックイン期間にチェックインしなかったことがわかった場合、そのエージェントは非アクティブなエージェントと宣言されます。

[最新のリビジョンへのソフトウェアエージェントの更新（Software Agent Updates to Latest Revision）] チャート

エージェントがコンフィギュレーションサーバーにチェックインするたびに、エージェントは現在の RPM バージョンも提示します。エージェントが特定のバージョンに設定されていて、2 回のチェックイン期間後に更新できていなかった場合、そのエージェントは最新バージョンにアップグレードできないと宣言されます。

[欠落エージェントパケット（Agent Packet Missed）] チャート

まれに、ホストを通過するトラフィック量がエージェントの検査できるレートよりも多い場合、一部のパケットが分析からスキップされます。欠落パケット数と対応するエージェント名がこのチャートに表示されます。

[エージェントのソフトウェアバージョン/OS分布（Agent Software Version/OS Distribution）] チャート

これらのグラフには、Secure Workload クラスタに登録されているすべてのエージェントのエージェントバージョン分布と親 OS プラットフォームが表示されます。

適用ステータス

適用ステータスを表示するには、ウィンドウの左側のナビゲーションバーの **[保護（Defend）]**> **[適用ステータス（Enforcement Status）]** をクリックします。

このページは、サイト管理者/カスタマーサポートユーザーと範囲所有者が、全適用エージェントの現在のステータス概要（ポリシーを適用しているクラウドコネクタを含む）を取得するために使用できます。

いずれかのチャートで赤またはオレンジが表示されている場合は、該当するトピックを参照してください。

表 *39* : 適用ステータスチャート

チャート（Chart）	結果	アクションの実行
適用が有効なエージェント（Agent Enforcement Enabled）	有効化されていない（Not Enabled）	エージェント設定で適用が有効になっていることを確認します。エージェント設定プロファイルの作成 （90 ページ）を参照してください。

チャート（**Chart**）	結果	アクションの実行
エージェントポリシーの設定（Agent Policy Config）	古いポリシー（Stale Policies）	一般的に、この状況は一時的なものであり、通常はアクションを必要としません。これは、ラベルに基づく Secure Workload 展開によってインベントリとポリシーが動的に更新されるために発生します。 ただし、個々のワークロードでこの状況が続く場合は、Cisco TAC にお問い合わせください。
エージェントの具体的なポリシー（Agent Concrete Policies）	スキップ（Skipped）	これは、ポリシーが一部のエージェントにプッシュされなかったことを示します。



ヒント

- 個々の範囲またはテナント全体のステータスを表示するには、ページの左上にある [範囲でフィルタ（Filter by Scope）] オプションを使用します。
- チャートに問題が示されている場合は、チャートの関連部分をクリックして、問題のあるワークロードを特定します。

  チャートの下のテーブルには、影響を受けるワークロードが表示されます。

  または、フィルタリングオプションを表示するには、チャートの下にある [フィルタ（Filter）] ボックス内の [(i)] ボタンをクリックします。
- 豊富な追加の詳細を表示するには、フィルタされたワークロードのリスト内で IP アドレスリンクをクリックして、[ワークロードプロファイル（Workload Profile）] ページを表示します。

次の表で、適用ステータステーブルに表示されるフィールドについて説明します。

フィールド	説明
[Host Name]	ワークロードのホスト名。
アドレス（Address）	ワークロード上のすべてのインターフェイスの IP アドレス。
[有効化された適用（Enforcement Enabled）]	エージェントで適用が有効になっているかどうかを示します。
[同期されている具体的なポリシー（Concrete Policies in Sync）]	必要なバージョンの具体的なポリシーが現在エージェントに適用されているかどうかを示します。

[具体的なポリシー（Concrete Polices）]	いずれかのホストでこの値が [スキップ（Skipped）] と表示されている場合は、そのホスト上のエージェントがポリシーの制限に達していることを意味します。（ポリシーに関連する制限 （1298 ページ） を参照。）
[ポリシー数（Policy Count）]	エージェントの具体的なポリシーの数。
ステータス（Status）	最新のポリシー設定適用のステータス。ステータスが [CONFIG_SUCCESS] の場合、現在のバージョンが問題なく適用されていることを示します。

クラウドコネクタの適用ステータス

AWS または Azure クラウドコネクタを設定している場合：

すべてのインターフェイスの適用ステータスを、適用ステータスページで確認できます。ポリシーが正常に適用された場合、ポリシーが同期していることがわかります。そうでない場合は、対応するエラーメッセージが表示されます。

適用ステータスページのポリシー数は Secure Workload アカウンティングで、AWS または Azure ルールアカウンティングではありません。

（AWS のみ）このページのホスト名フィールドは、パブリック DNS から取得されます。指定された VPC でパブリック DNS が有効になっていない場合、ホスト名フィールドは空になります。

ポリシー更新の一時停止

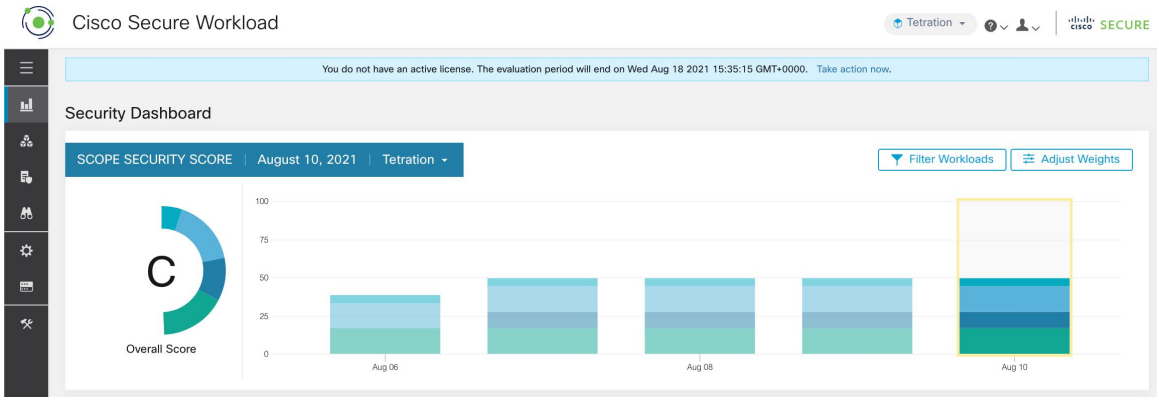
ポリシー更新の一時停止 （672 ページ） を参照してください。

ライセンス

Secure Workload ライセンスのステータスを表示するには、ウィンドウの左側にあるナビゲーションバーで、**[管理（Manage）]** > **[ライセンス（Licenses）]** をクリックします。

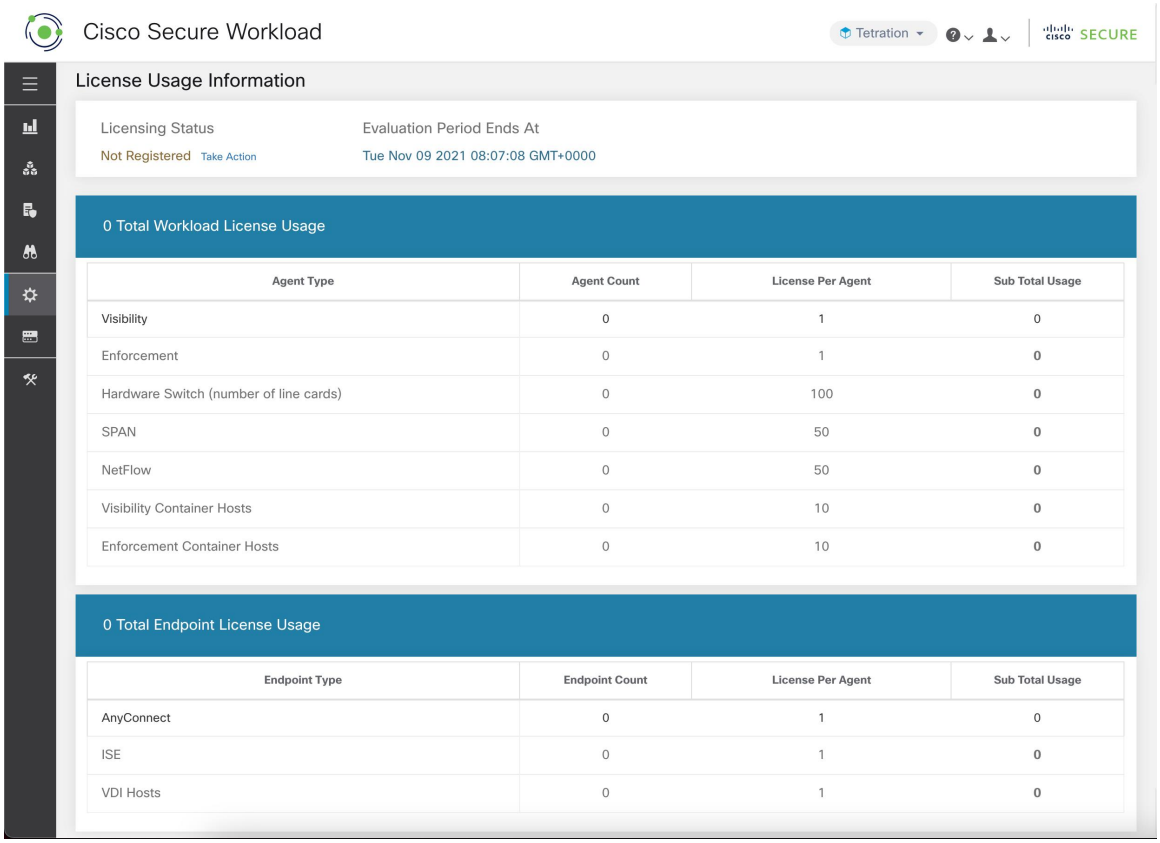
このページは、サイト管理者が現在のライセンスステータスとライセンス使用状況の概要を取得するために使用します。このリリース以降では、オンプレミスでの展開用にクラスタを登録する必要があります。このリリースで新しいクラスタにアップグレードするか、新しいクラスタを展開すると、ソフトウェアは自動的に 90 日間の評価モードに入ります。バナーが表示され、評価の有効期限が示されます。

図 *511* : ライセンスバナー



（注） 90 日以内に登録が正常に完了しなかった場合、バナーメッセージはコンプライアンス不適合に変わります。登録がないため、ブロックされる機能はありません。

図 *512* : *[*監視（*monitoring*）*]* - *[*ライセンス（*licenses*）*]* ページで表示される詳細なライセンス情報

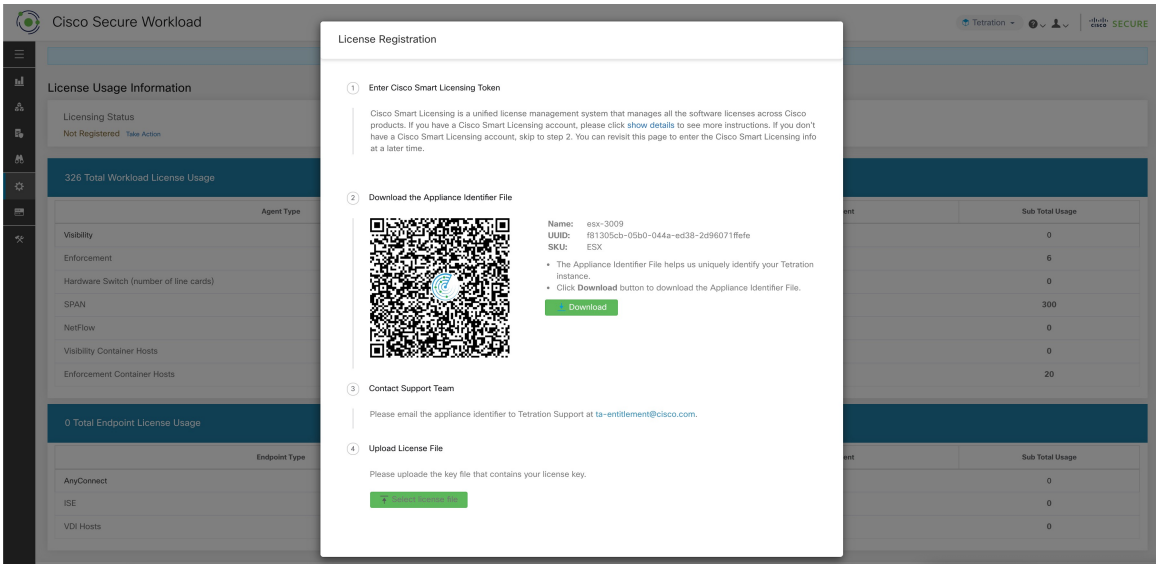


ライセンス登録

このセクションでは、ライセンスの取得方法について説明します。

ライセンスバナーまたは **[管理（Manage）]** > **[ライセンス（Licenses）]** ページで [アクションの実行（Take Action）] をクリックして、ライセンスを要求します。クラスタ識別ファイルをダウンロードする方法とライセンスを取得する方法についての説明が表示されます。

図 *513*：ライセンス登録モーダル：クラスタ識別ファイルのダウンロード



手順

ステップ 1 **ライセンス登録モーダル**を完了するには、CSSM スマート ソフトウェア ライセンス ポータルで生成された登録トークンが必要です。CSSM を使用してトークンを生成する手順は、ライセンスモーダル自体に表示されます。登録トークンを取得したら、トークンをコピーしてライセンスモーダルのテキストボックスに貼り付け、テキストボックスの横にある [送信（Submit）] ボタンをクリックします。

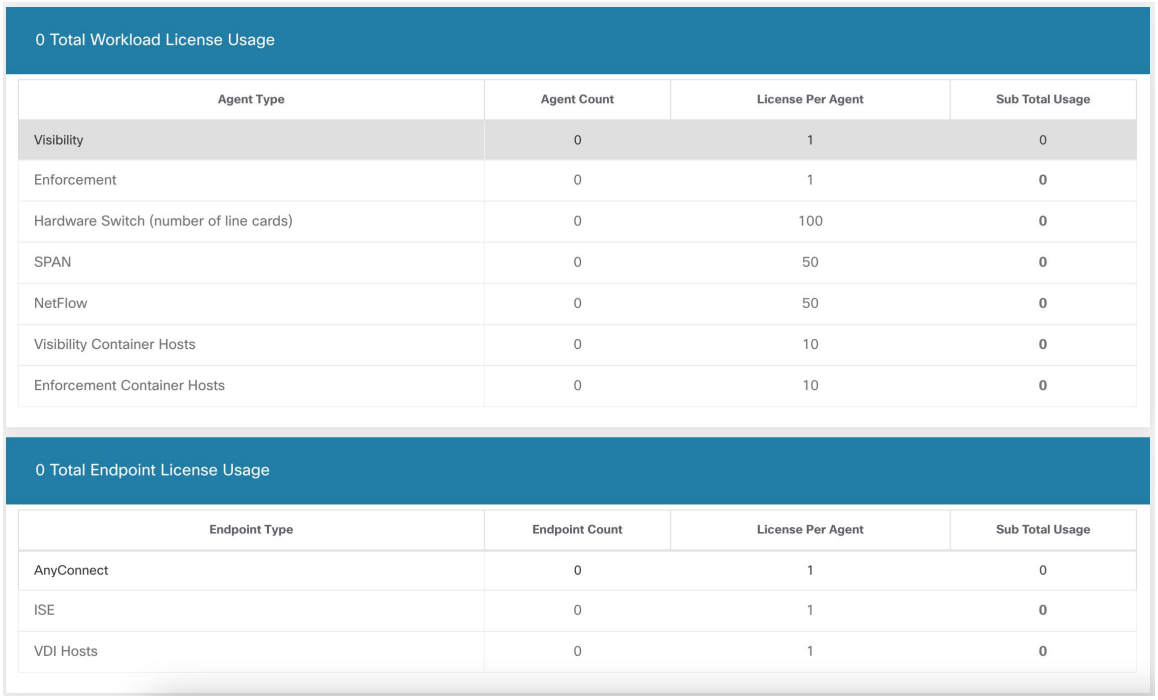
ステップ 2 次に、[Download] ボタンをクリックして、クラスタ識別ファイルをローカルストレージにダウンロードします。識別ファイルのファイル名形式は **reg_id_<cluster_name> _<cluster_uuid> .gz** です。ID ファイルには、IP アドレス情報、特定のワークロードの詳細、または PII 情報は含まれていません。この ID ファイルを ta-entitlement@cisco.com に送信する必要があります。**ライセンスキーファイル**を含む応答が、識別ファイルを受信したときと同じ電子メールアドレスに送信されます。

ステップ 3 この**ライセンスキーファイル**をライセンスモーダルからアップロードする必要があります。応答ファイルをアップロードするには、ライセンスモーダルの手順 4 を使用します。

ライセンス使用状況の確認

このセクションでは、詳細なライセンス使用状況を確認する方法について説明します。左側のナビゲーションバーで、**[管理（Manage）]** > **[ライセンス（Licenses）]** をクリックします。

図 *514*：ライセンステーブルと詳しい使用状況



0 Total Workload License Usage

Agent Type	Agent Count	License Per Agent	Sub Total Usage
Visibility	0	1	0
Enforcement	0	1	0
Hardware Switch (number of line cards)	0	100	0
SPAN	0	50	0
NetFlow	0	50	0
Visibility Container Hosts	0	10	0
Enforcement Container Hosts	0	10	0

0 Total Endpoint License Usage

Endpoint Type	Endpoint Count	License Per Agent	Sub Total Usage
AnyConnect	0	1	0
ISE	0	1	0
VDI Hosts	0	1	0

（注） 登録後、ライセンス使用量がエンタイトルメント（ワークロードまたはエンドポイント）を超えると、非準拠の警告バナーが UI に表示されます。ライセンス使用量を上回っても、追加センサーのインストールを含め、どの機能もブロックされることはありません。使用量がエンタイトルメントを下回ると、準拠に関する警告バナーは消えます。追加のライセンスを購入した場合は、ID 情報（ライセンスモーダルから再度ダウンロード）とともに ta-entitlement@cisco.com に連絡して、更新されたライセンスキーファイルを要求できます。

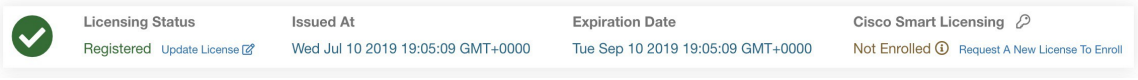
Cisco Smart Licensing の詳細

Cisco スマートライセンスは統合ライセンス管理システムであり、Cisco 製品のソフトウェアライセンスすべてを管理します。Cisco Smart Licensing アカウントを持っている場合は、Cisco Smart Licensing Token を Secure Workload ライセンスに関連付けることができます。Cisco Smart Licensing アカウントを持っていない場合は、Cisco Smart Licensing なしでライセンスを取得または更新できます。

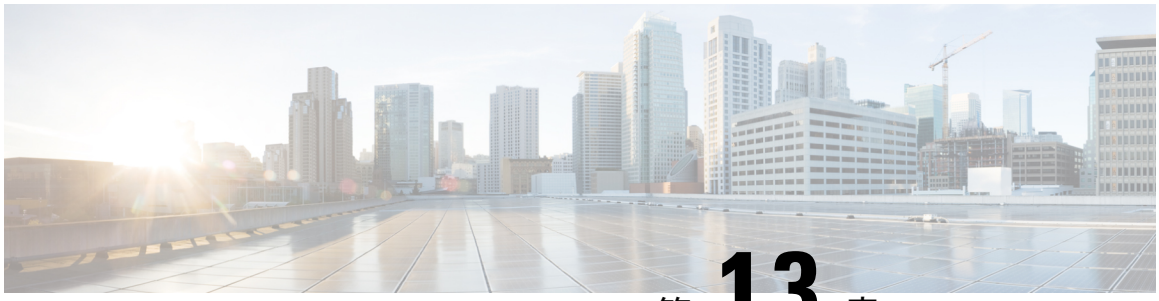
手順

ステップ 1 有効な Secure Workload ライセンスをすでに持っている場合は、[登録する新しいライセンスをリクエスト（Request A New License To Enroll）] をクリックして、Cisco Smart Licensing Token を使用して新しいライセンスを取得できます。

図 *515 :* 新しいライセンスを取得して、*Cisco Smart Licensing Token* をライセンスに *Secure Workload* に関連付ける



ステップ 2 有効な Secure Workload ライセンスがない場合は、前のセクションの説明に従い、[アクションの実行（Take Action）] をクリックして新しいライセンスを取得できます。



第 13 章

脅威インテリジェンス

脅威インテリジェンスを管理するには、左側のナビゲーションバーで **[管理（Manage）] > [脅威インテリジェンス（Threat Intelligence）]** をクリックします。

[脅威インテリジェンス（Threat Intelligence）] 機能セットは、Secure Workload パイプラインに最新のデータセットを提供します。このパイプラインは、外部の既知のマルウェア コマンド アンド コントロール アドレス、プロセスのセキュリティフローと地理的位置に関してデータセンターのワークロードを検査することにより、脅威を識別および検疫します。

脅威インテリジェンス ダッシュボードには、脅威インテリジェンス データセットの最新の更新ステータスが表示されます。これらのデータセットは自動的に更新されます。

警告 脅威インテリジェンス機能を自動更新するには、Cisco Secure Workload サーバーへの接続が必要です。エンタープライズアウトバウンドHTTPリクエストには、次が必要になる場合があります。

1. エンタープライズファイアウォールのアウトバウンドルールから次のドメインを許可します。
   - uas.tetrationcloud.com
2. アウトバウンド HTTP 接続を設定します。

アウトバウンド接続のない環境では、これらのデータセットを直接アップロードできます。「手動アップロード」の項を参照してください。

データセット

データセット	説明
NVD CVE	セキュリティ関連のソフトウェアの欠陥、CVSSベーススコア、脆弱な製品設定、および弱点の分類
MaxMind Geo	送信元 IP の場所および他の特性の特定

データセット	説明
NIST RDS	既知の追跡可能なソフトウェアアプリケーションのデジタル署名の NIST 参照データセット
Team Cymru	3,000 を超えるボットネット コマンド アンド コントロール IP に関する知見
ハッシュ判定	プロセスハッシュに関する Secure Workload の判定（[自動更新（Automatic Updates）] セクションでのみ利用可能）。

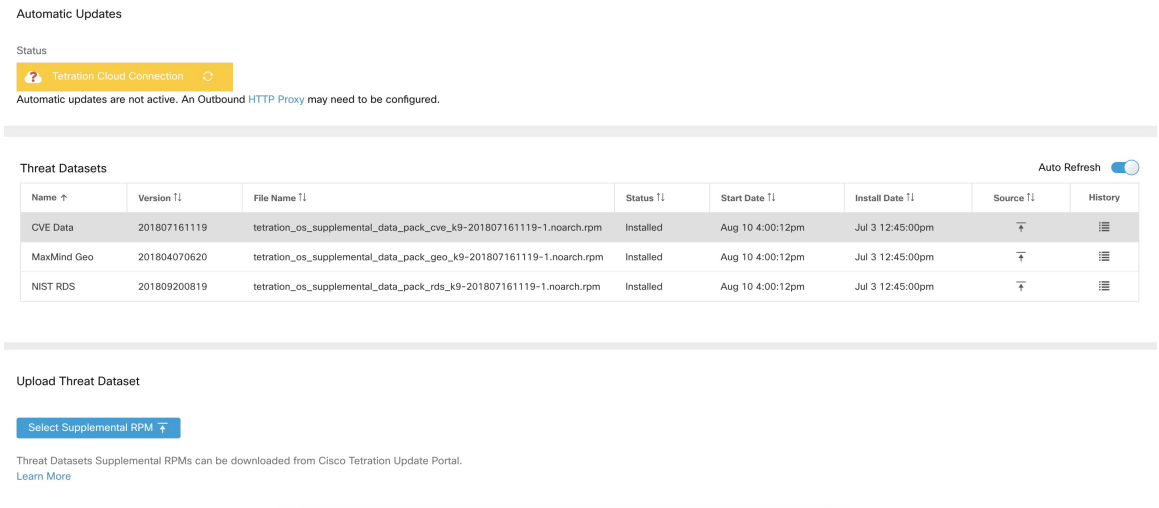
（注） MaxMind Geo データセットが以前のリリースで手動でアップロードされた場合は、対応する RPM を再アップロードして、フロー可視性のページで場所と関連情報を表示してください。

- 自動更新 （932 ページ）
- 手動アップロード （933 ページ）

自動更新

脅威データセットの更新はアプライアンスからトリガーされ、毎日午前 3 時から 4 時（UTC）に、uas.tetrationcloud.com のインターネット上でホストされているグローバルデータセットで入手可能なグローバルデータセットと同期されます。グローバルデータセットは、毎週金曜日または月曜日に更新されます。脅威インテリジェンス ダッシュボードには、データセットとデータセットの最終更新日が一覧表示されます。

図 *516* : ダッシュボード



手動アップロード



注目 **手動アップロードのスケジューリング** データセット rpm ファイルは、毎週 Cisco Secure Workload Update Portal に公開されます。最新のリリースを定期的にインストールし、管理者がインストールするようにスケジュールを設定することを推奨します。

更新されたデータセットのダウンロード

データセットは Secure Workload 更新ポータルからダウンロードできます。

Cisco Secure Workload へのアップロード

このセクションでは、データセット rpm ファイルをアップロードする方法について説明します。

始める前に

[サイト管理者（Site Admin）] または [カスタマーサポート（Customer Support）] としてシステムにログインする必要があります。

手順

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）] > [脅威インテリジェンス（Threat Intelligence）]** をクリックします。

ステップ 2 [脅威データセットのアップロード（Upload Threat Dataset）] セクションまでスクロールします。

ステップ 3 [補足RPMを選択（Select Supplemental RPM）] をクリックします。

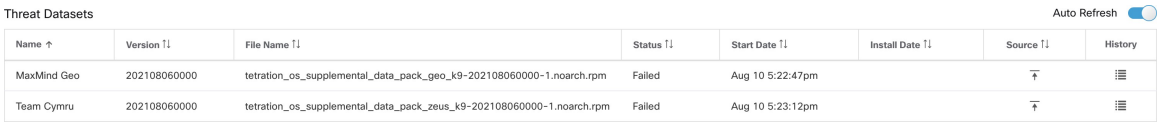
ステップ 4 Secure Workload 更新ポータルからダウンロードした rpm ファイルを選択します。

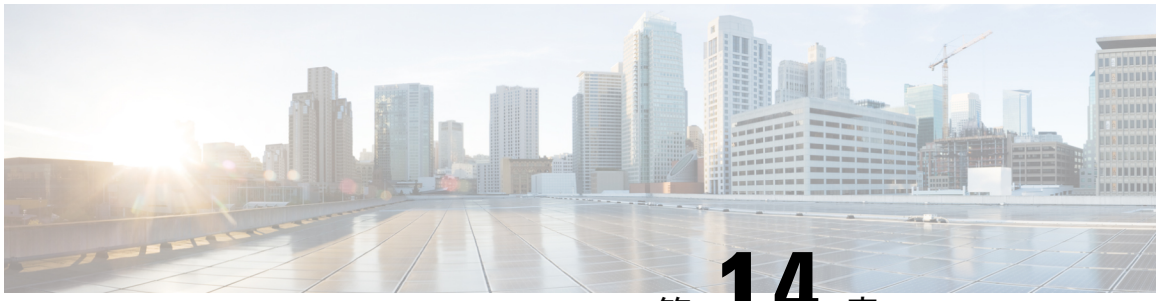
ステップ 5 準備ができると、確認ダイアログが表示されます。[アップロード（Upload）] をクリックします。

ステップ 6 rpm がアップロードされます。進行状況バーが表示されます。アップロードが完了すると、ダイアログが閉じます。

ステップ 7 その後、rpm が処理され、バックグラウンドでインストールされます。インストールが完了すると、テーブルが更新されます。

図 *517 :* テーブルの更新





第 14 章

レポートダッシュボード

レポートダッシュボードは、エグゼクティブ、ネットワーク管理者、およびセキュリティアナリスト向けに設計されており、重要なワークフローステータスの視覚表示、トラブルシューティング機能、およびレポート作成機能を提供します。

[レポートダッシュボード（Reporting Dashboard）] ページを使用するには、UI の [レポート（Reporting）] に移動します。以下の項では、レポートの概要と、レポートのスケジュール設定と電子メール送信の方法について説明します。

- レポートダッシュボード （935 ページ）

レポートダッシュボード

レポートダッシュボードは、エグゼクティブ、ネットワーク管理者、およびセキュリティアナリスト向けに設計されており、重要なワークフローステータスの視覚表示、トラブルシューティング機能、およびレポート作成機能を提供します。

[レポートダッシュボード（Reporting Dashboard）] ページを使用するには、UI の [レポート（Reporting）] に移動します。以下の項では、レポートの概要と、レポートのスケジュール設定と電子メール送信の方法について説明します。

概要

概要セクションでは、ネットワークフロー情報、セキュリティポリシー、システムパフォーマンス、およびセキュリティ脅威に関するリアルタイムのインサイトが提供されます。これにより、セキュリティアナリストとネットワーク管理者は、情報に基づいた意思決定を行い、データに基づいてリソースを保護するためのプロアクティブな対策を講じることができます。

セグメンテーションの概要

ワークスペースは、クラスタ内のポリシーと適用を検出、適用、および管理するための構成要素です。ユーザーは、適切な範囲を選択してセグメンテーションのメンバーシップを定義します。

セグメンテーションの概要では、導入されている各ワークスペースの設定の詳細がキャプチャされます。ワークスペース内の特定の範囲、またはその範囲に関連付けられたワークスペースのポリシーの定義、分析、適用など、すべてのポリシー関連のアクティビティが実行されます。

次のグラフは、ワークスペースに関連付けられているさまざまなポリシーの概要を示しています。

図 *518* : セグメンテーションの概要

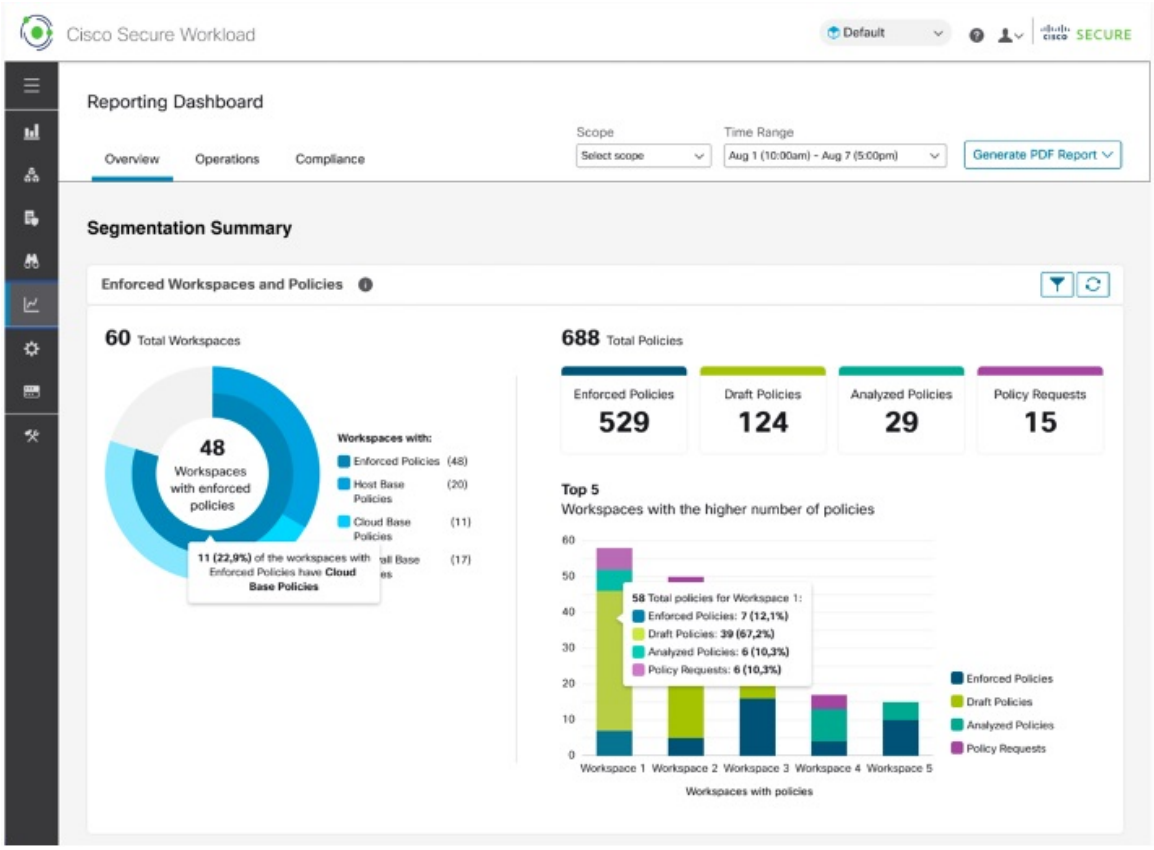
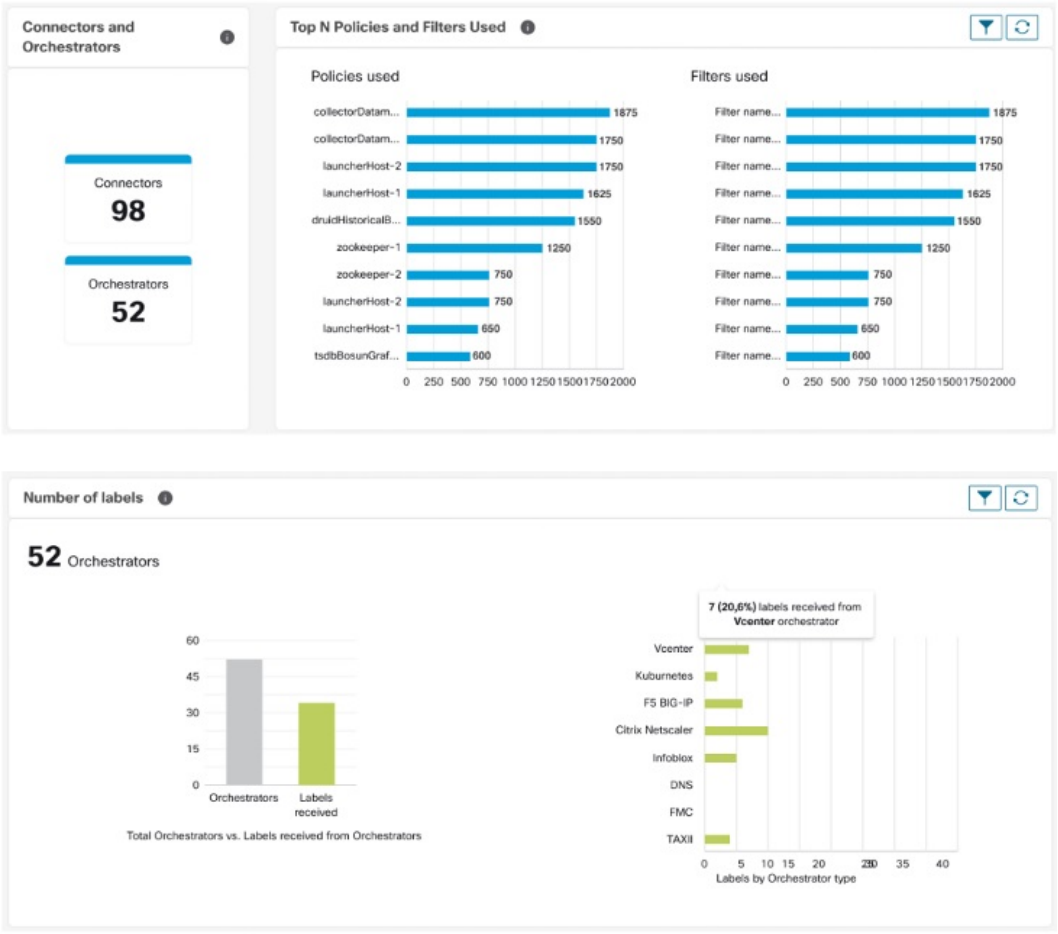


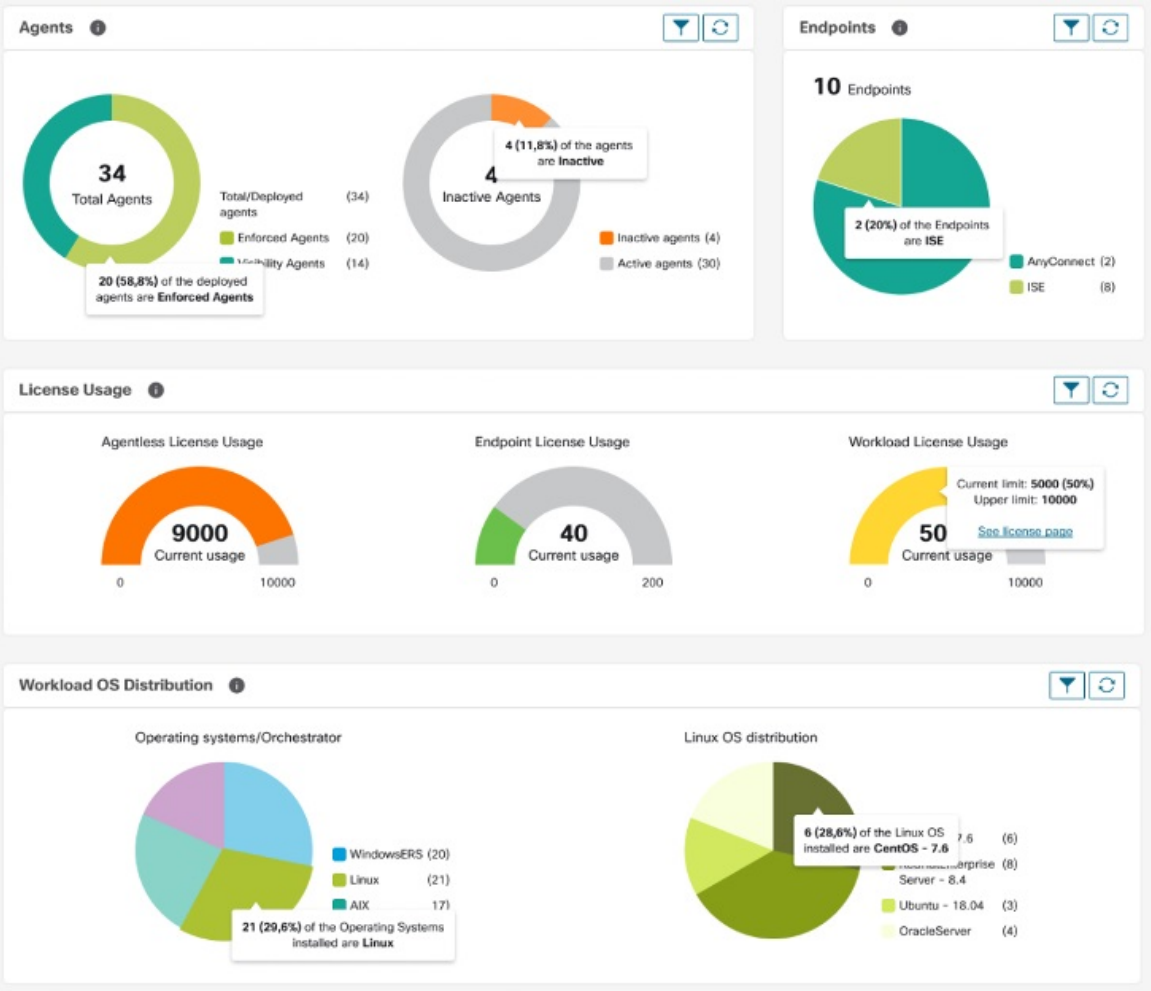
図 *519* : コネクタとオーケストレータ



ワークロードの概要

ワークロードの概要では、インフラストラクチャ内のサーバーとエンドポイントに展開されているエージェントの合計数の詳細が提供されます。エージェントは、ネットワークフロー情報をモニターおよび収集し、インストールされたホストにファイアウォールルールを使用してセキュリティポリシーを適用し、ワークロードのステータスを伝達し、セキュリティポリシーの更新を受信します。

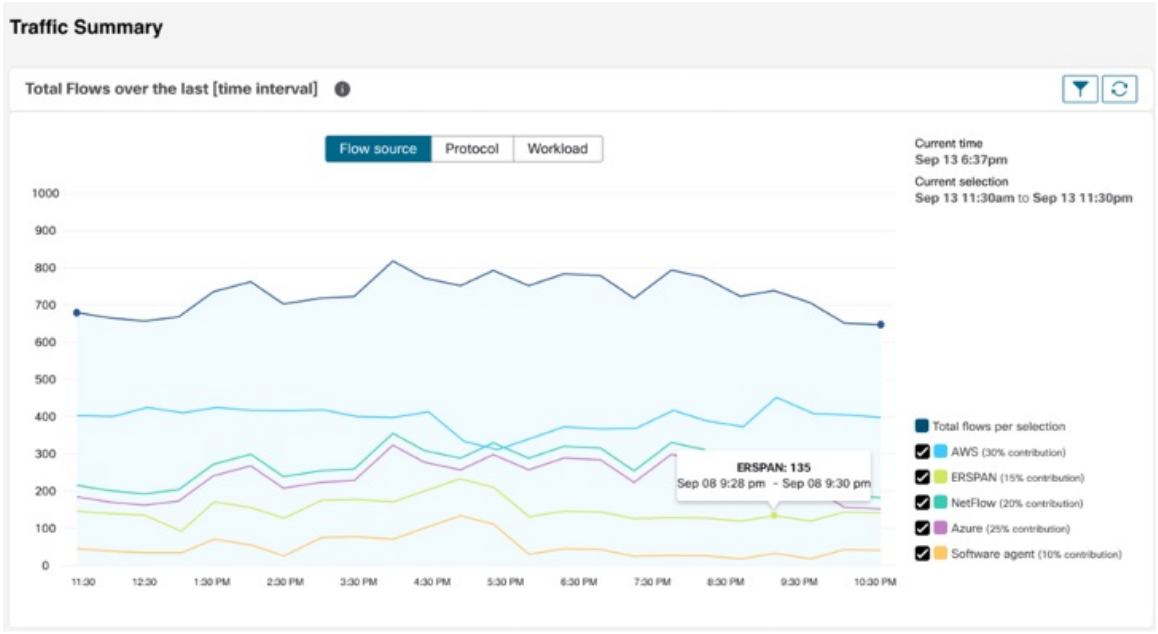
図 *520 :* ワークロードの概要



Traffic Summary

トラフィックの概要では、各フローのフロー観測がキャプチャされます。フローソースの各観測で、フローのパケット数、バイト数、およびその他のメトリックが追跡されます。

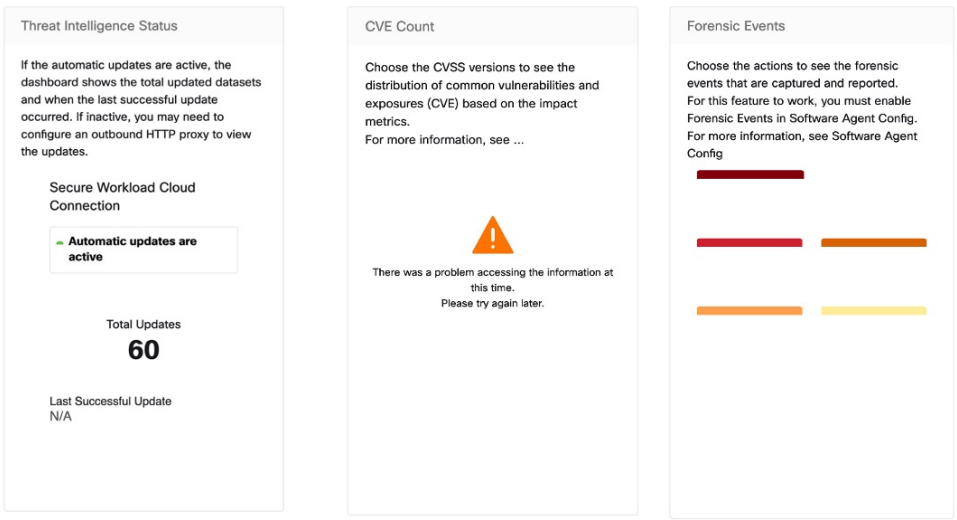
図 *521 : Traffic Summary*



セキュリティ サマリー

セキュリティの概要には、脅威インテリジェンスのステータス（脅威インテリジェンスのステータスの更新を最後に受信した時刻が表示されます）、CVEの数、およびフォレンジックイベントの分布が表示されます。

図 *522 :* セキュリティ サマリー

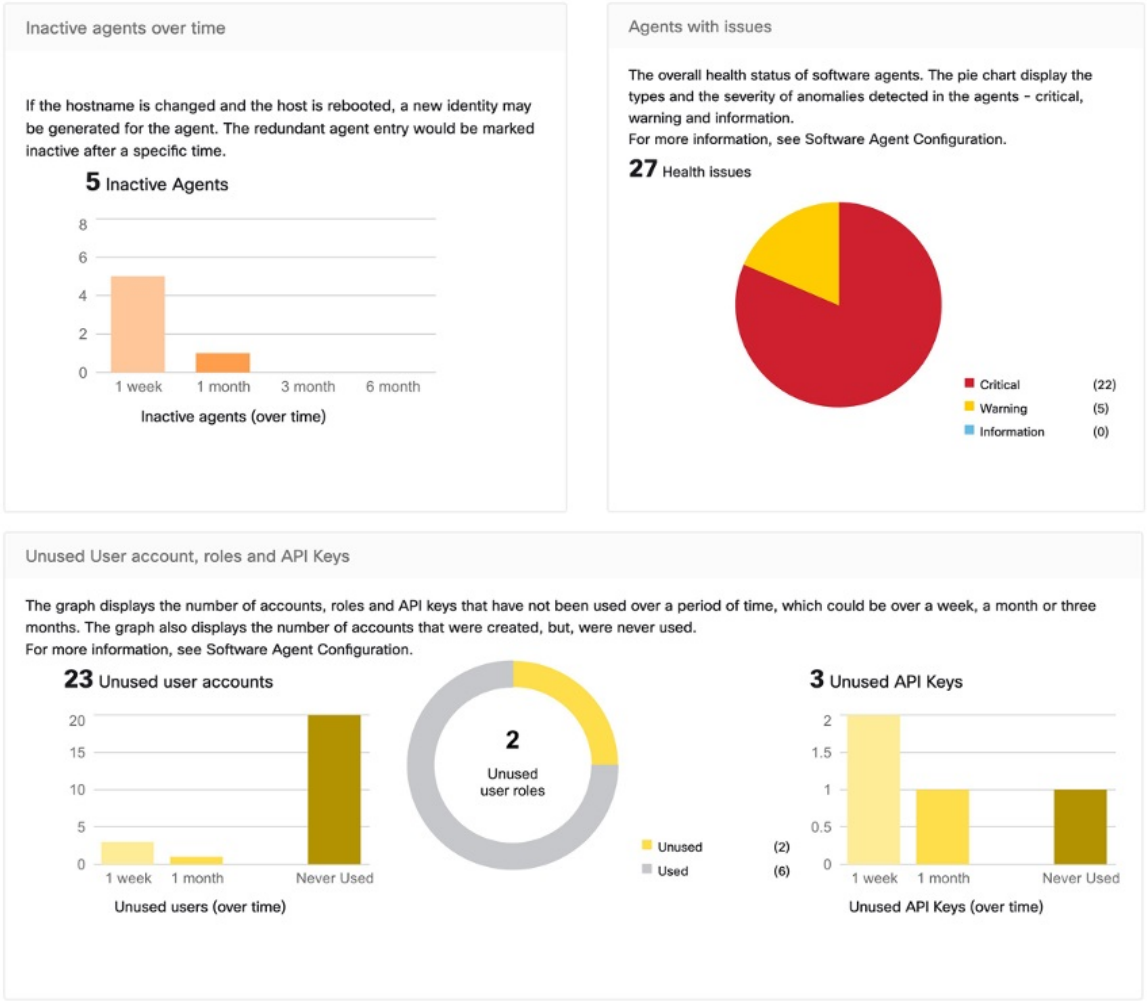


動作

ワークロードの概要

ワークロードの概要には、ネットワーク内のサーバーとエンドポイントに展開されているエージェントの合計数が表示されます。エージェントは、ネットワークフロー情報をモニターおよび収集し、インストールされたホストにファイアウォールルールを使用してセキュリティポリシーを適用し、ワークロードのステータスを伝達し、セキュリティポリシーの更新を受信します。

図 *523 :* ワークロードの概要

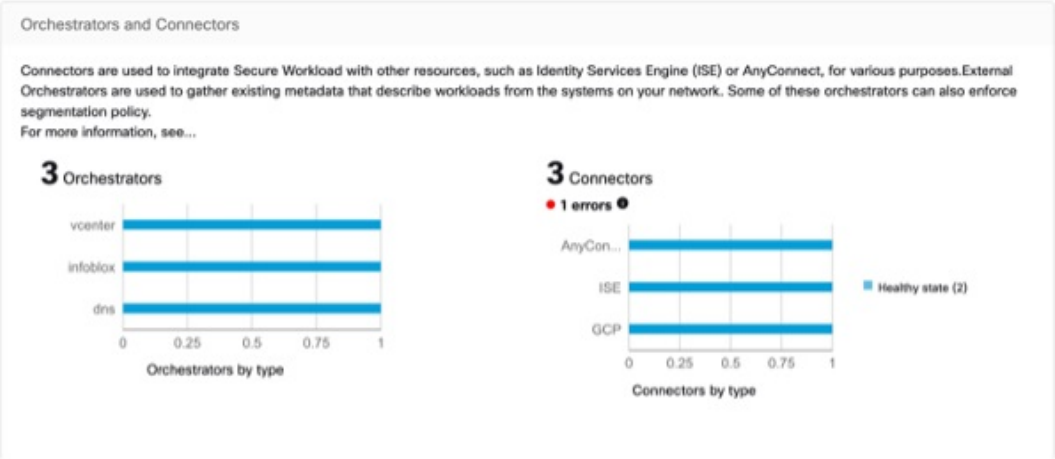


テレメトリの概要

仮想アプライアンスに展開された多くのコネクタは、ネットワーク内のさまざまなポイントからテレメトリを収集します。これらのコネクタは、アプライアンスの特定のポートでリッスンする必要があります。特定のセキュリティグループのフローログを設定している場合、コネク

タはフローログを取り込むことができます。また、可視化およびセグメンテーションポリシーの生成にテレメトリデータを使用することもできます。

図 *524* : テレメトリの概要



クラスタ要約

サイト管理者はクラスタステータスページにアクセスできますが、アクションはカスタマーサポートユーザーのみが実行できます。Cisco Secure Workload ラック内にあるすべての物理サーバーのステータスが表示されます。

クラスタの処理時間と保持時間は、クラスタ内でデータが保存および処理される期間を指します。具体的な処理時間と保持時間は、ワークロードの要件と組織のポリシーによって異なります。

クラスタを設定する際は、処理時間の要件を考慮することが重要です。これは、ワークロードのニーズを満たすために必要なストレージ容量と処理能力に影響を与える可能性があるためです。

保持時間は、データがクラスタ内に保持される時間の長さを指します。一部のワークロードでは規制またはコンプライアンスに関連する目的のためにデータを保持する必要が生じる場合がありますが、他のワークロードでは処理後に削除される場合もあります。ワークロードの保持ポリシーを確立して、データを適切な期間保持し、その後安全に削除して不正アクセスを防ぐことが重要です。

図 *525 :* クラスタ要約



セグメンテーションの概要

セグメンテーションまたはアプリケーションワークスペースは、クラスタ内のポリシーと適用を検出、適用、および管理するための構成要素です。セグメンテーションの概要では、導入されている各アプリケーションワークスペースの設定の詳細がキャプチャされます。これには、適用があるワークスペースとないワークスペースの数、有効化/無効化されたポリシーの数、最新のポリシーがあるワークスペースまたは同期されていないワークスペースの数、およびドラフトポリシーがあるワークスペースとないワークスペースの数が含まれています。

図 *526 :* セグメンテーションの概要



コンプライアンス

ワークロードの概要

ワークロードの概要には、インフラストラクチャ内のサーバーとエンドポイントに展開されているエージェントの合計数が表示されます。エージェントは、ネットワークフロー情報をモニターおよび収集し、インストールされたホストにファイアウォールルールを使用してセキュリティポリシーを適用し、ワークロードのステータスを伝達し、セキュリティポリシーの更新を受信します。

図 *527 :*



セキュリティ サマリー

フォレンジックイベントを設定します。設定すると、すべての戦術がその下のルールなし、カウント 0 の状態で表示されます。戦術レベルで選択するには、フォレンジックルールを 1 つ以上選択します。戦術を選択すると、その下にあるすべてのルールが選択されます。デフォルトの MITRE ATT&CK ルールは、MITRE ATT&CK フレームワークからのアラート手法に対して指定されています。

図 *528 :*



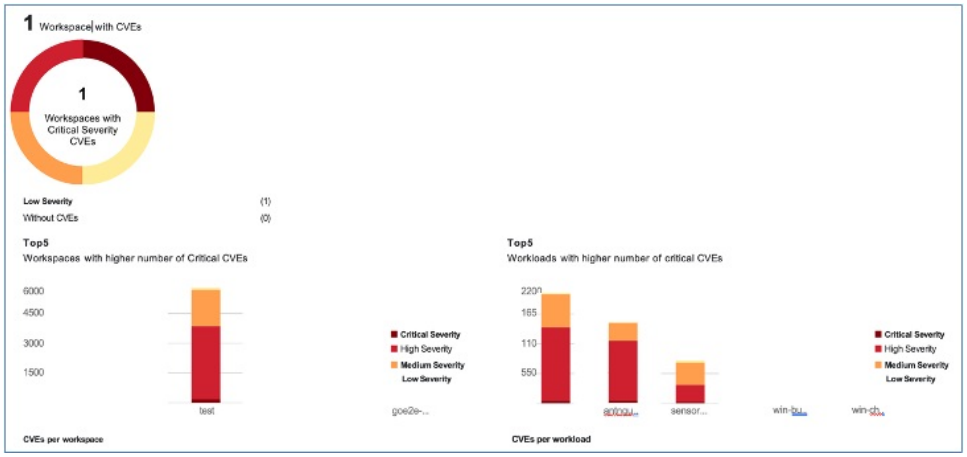
CVE を含むワークスペース

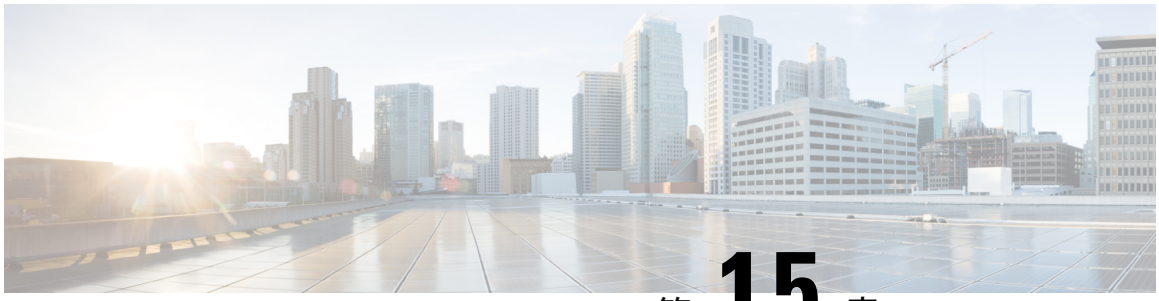
CVE のカウントでは、選択した範囲とスコアリングシステム（v2 または v3）に基づいて、選択した範囲内のワークロードの脆弱性（スコア順に並べ替え）が強調表示されます。重大な CVE の数が最も多いワークスペースとワークロードの分布を確認します。

ワークロード上のソフトウェアパッケージは、既知の脆弱性（CVE）に関連付けられている可能性があります。共通脆弱性評価システム（CVSS）は、CVE の影響を評価するために使用されます。CVE には、CVSS v2 および CVSS v3 スコアを設定できます。脆弱性スコアを計算するために、使用可能な場合は CVSS v3 が考慮され、それ以外の場合は CVSS v2 が考慮されます。

ワークロードの脆弱性スコアは、そのワークロードで検出された脆弱なソフトウェアのスコアから導き出されます。ワークロード脆弱性スコアは、CVSS スコアおよびベンダーデータに基づいて計算され、データが欠落しているか不正確な場合、セキュリティ調査チームによって調整される場合があります。最も重大な脆弱性の重大度が高いほど、スコアは低くなります。

図 *529 :*


1 Workspace with CVEs
1
Workspaces with Critical Severity CVEs
Low Severity (1)
Without CVEs (0)
Top5
Workspaces with higher number of Critical CVEs
6000
4500
3000
1500
test
goe2e-...
Critical Severity
High Severity
Medium Severity
Low Severity
CVEs per workspace
Top5
Workloads with higher number of critical CVEs
220
165
110
550
sensor...
win-bl...
win-cb...
Critical Severity
High Severity
Medium Severity
Low Severity
CVEs per workload



第 15 章

セキュリティ ダッシュボード

セキュリティダッシュボードは、Cisco Secure Workload で利用可能な複数のシグナルをまとめることで、実用的なセキュリティスコアを提示します。現在のセキュリティポジションを把握し、改善するのに役立ちます。

セキュリティダッシュボードは、フロー検索、インベントリ検索、自動ポリシー検出、フォレンジックなど、Cisco Secure Workload 内での多くの詳細なドリルダウンの出発点として機能します。

• セキュリティダッシュボードへの移動 （948 ページ）
• セキュリティスコア （948 ページ）
• セキュリティスコアのカテゴリ （948 ページ）
• ハイレベルビュー （949 ページ）
• 範囲レベルスコアの詳細 （949 ページ）
• スコアの詳細 （951 ページ）

セキュリティダッシュボードへの移動

セキュリティダッシュボードを表示するには、ウィンドウの左側にあるナビゲーションバーで [概要（Overview）] をクリックします。

セキュリティスコア

セキュリティスコアは、0～100の数値で、カテゴリ内のセキュリティポジションを示します。100 のスコアが最高のスコアで、0 のスコアが最悪です。スコアは 100 に近いほど良くなります。

セキュリティスコアの計算では、インストールされているソフトウェアパッケージの脆弱性、プロセスハッシュの一貫性、さまざまなインターフェイスの開いているポート、フォレンジックおよびネットワーク異常イベント、ポリシーへの準拠/非準拠が考慮されます。

セキュリティスコアのカテゴリ

6 種類のスコアカテゴリがあります。ワークロードのセキュリティ面の大部分が、これらのカテゴリを導出するために考慮されます。

- [脆弱性スコア（Vulnerability Score）]：ワークロードにインストールされているパッケージの脆弱性がスコアリングに使用されます。
- [プロセスハッシュスコア（Process Hash Score）]：プロセスハッシュの一貫性（および異常）に加え、無害なプロセスハッシュおよびフラグ付きプロセスハッシュがスコアリングに使用されます。
- [攻撃対象スコア（Attack Surface Score）]：プロセスでは、サービスを利用可能にするために、複数のインターフェイスで1つ以上のポートが開いている場合があります。未使用のオープンポートが、スコアリングに使用されます。
- [フォレンジックスコア（Forensics Score）]：ワークロードのフォレンジックイベントの重大度がスコアリングに使用されます。
- [ネットワーク異常スコア（Network Anomaly Score）]：ワークロードのネットワーク異常イベントの重大度がスコアリングに使用されます。

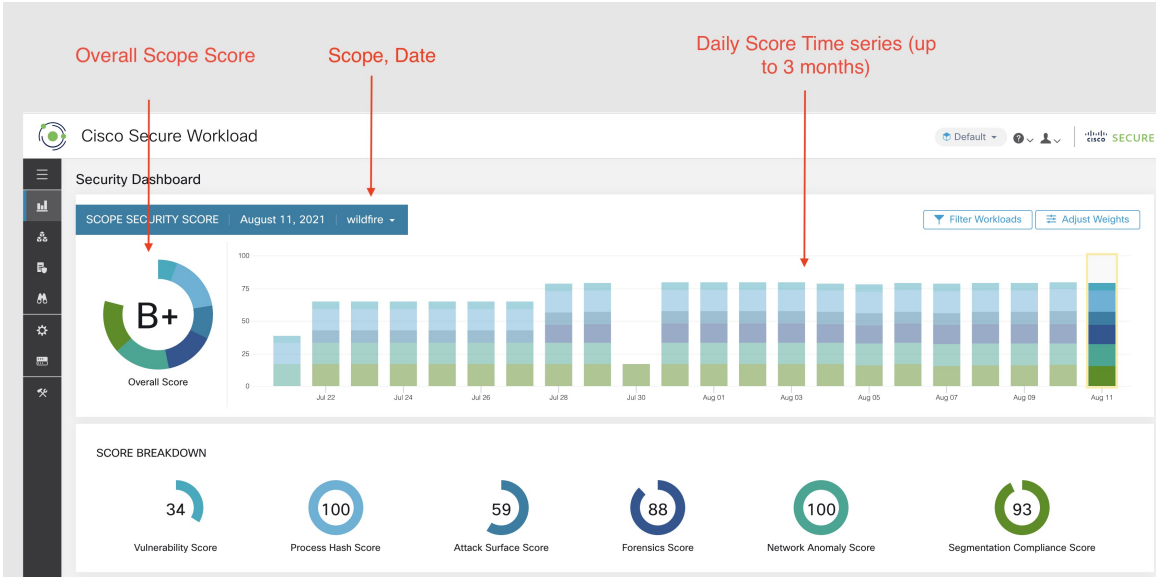
- [セグメンテーションコンプライアンススコア（Segmentation Compliance Score）]：自動的に検出されたポリシーに対する遵守（許可）および違反（エスケープ）がスコアリングに使用されます。

ハイレベルビュー

セキュリティダッシュボードには、選択した範囲の範囲レベルスコアが表示されます。時系列の総合スコアとスコアの内訳があり、選択した範囲の 6 つのスコアカテゴリのスコア詳細が 1 つずつ下に表示されます。

範囲レベルスコアの詳細

範囲レベルスコアの詳細はダッシュボードの上部にあります。



以下の内容が表示されます。

- **全体的な範囲スコア**：選択した範囲の全体的なスコア。
- **日次スコア時系列**：3 か月までの積み上げ時系列。
- **スコアの内訳**：時系列での選択した日のカテゴリスコアの内訳。

総合スコア

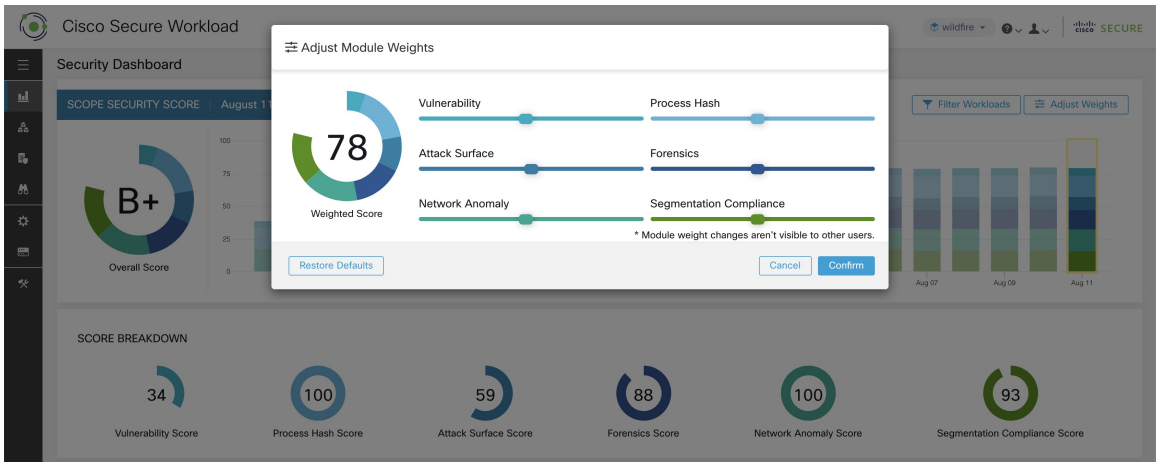
総合スコアは、**A+**、**A ～ F** の文字です。**A+** が最も良いスコアです。**F** が最も悪いスコアです。これは、スコアカテゴリを表す各スライス（色分け）で構成されるドーナツグラフです。



総合スコアは、スコアの6つのカテゴリの加重平均です。デフォルトでは、すべての重みは等しくなっています。スコアが **N/A** の場合、全体のスコア計算では 0 と見なされます。

$$Overall\ score = \frac{\sum W_{category} \times Score_{category}}{\sum W_{category}}$$

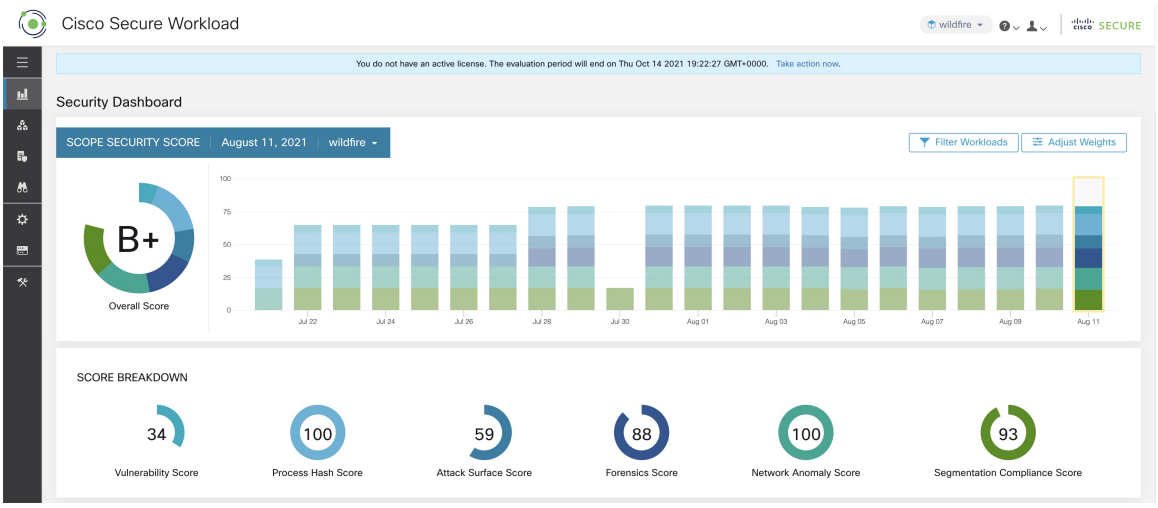
重みは、**重みの調整**モジュールのスライドを使用して調整できます。ユーザーごとに独自の重み調整を設定できるため、スコアをユーザーの優先順位に合わせることができます。



重要：スコアが **N/A** の場合、全体のスコア計算では **0** と見なされます。

日次時系列

最大3か月までの積み上げ時系列です。長期間にわたるセキュリティポジションの追跡に役立ちます。各スタックは、1 日の全体的なスコアを表します。スタック内の各セグメントは、異なる色で表されるカテゴリです。日をクリックすると、その日のスコアの内訳が表示されます。



スコアの内訳

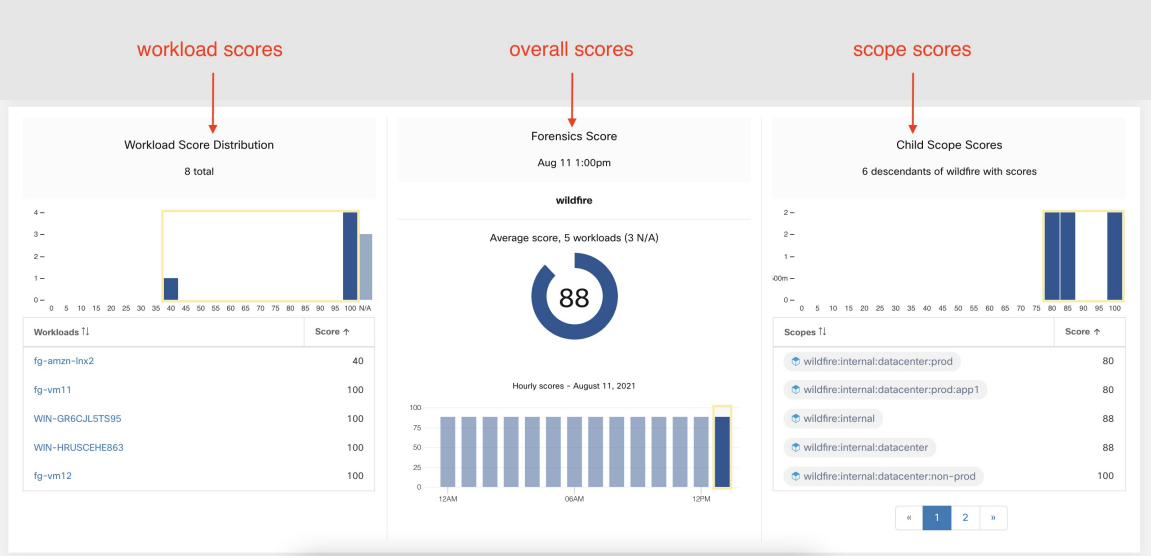
[Score Breakdown（スコアの内訳）] には、時系列で選択された日の 6 つのカテゴリすべてのスコアが表示されます。スコア **N/A** は、スコアが利用できないことを示します。全体のスコア計算では 0 としてカウントされます。



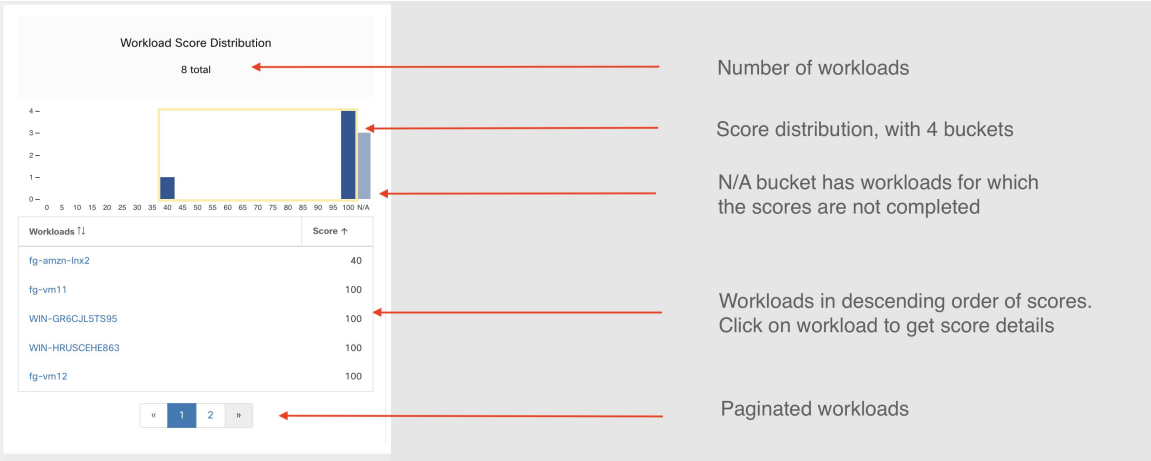
重要：スコアが **N/A** の場合、全体のスコア計算では **0** と見なされます。

スコアの詳細

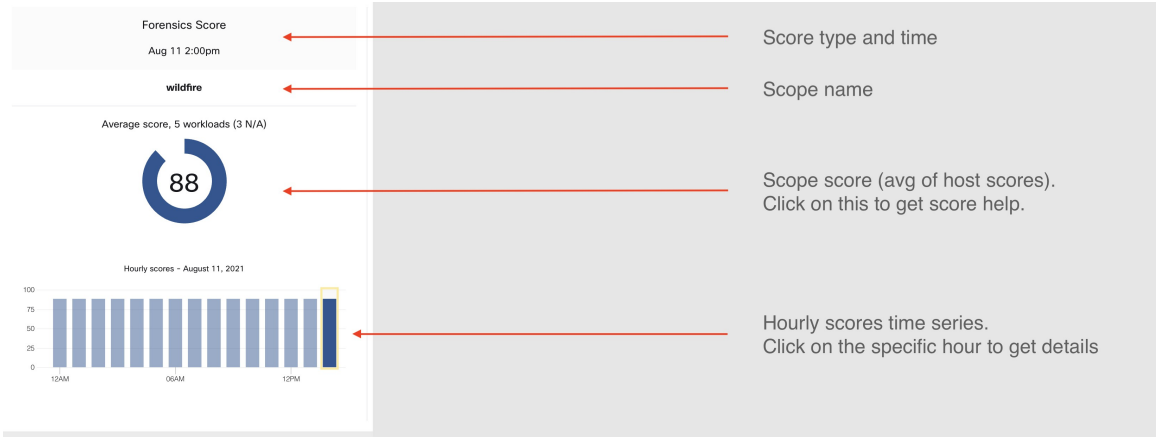
6 つのカテゴリは、それぞれ次のテンプレートに従います。テンプレートには、ワークロードスコアの分布、1 時間ごとの時系列、および子範囲のスコア分布があります。



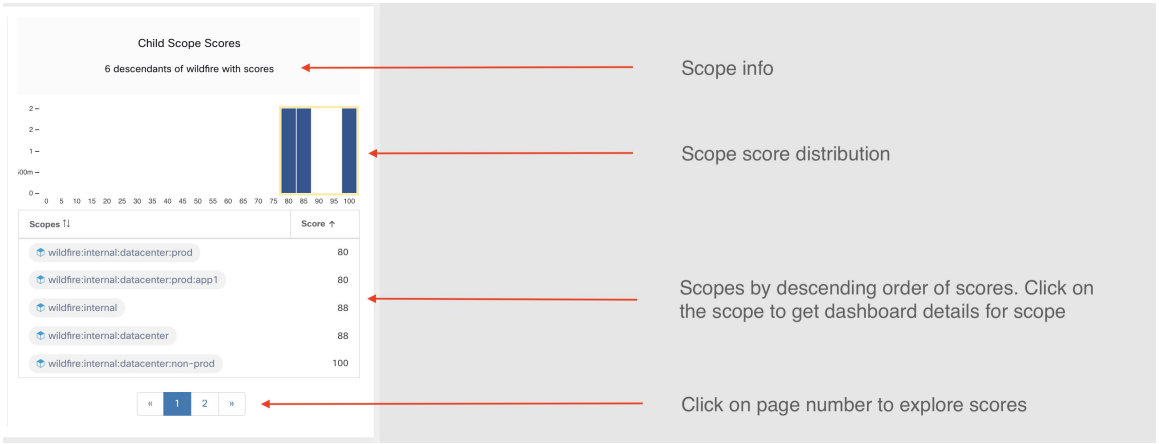
ワークロードスコアの分布では、選択した範囲の下のワークロードからのスコアの貢献度についての分析情報が得られます。分析情報により、最も低いスコアのワークロードをバブルアップして修正措置を迅速化するのに役立ちます。



1時間ごとの時系列は、選択した1日を通して1時間ごとのスコアを取得するのに役立ちます。1時間ごとの時系列で時間を選択すると、ワークロードスコアの分布と子孫範囲の分布が更新され、選択した時間が表示されます。



子孫範囲の分布では、選択した範囲の子範囲のスコア貢献度に関する分析情報が得られます。

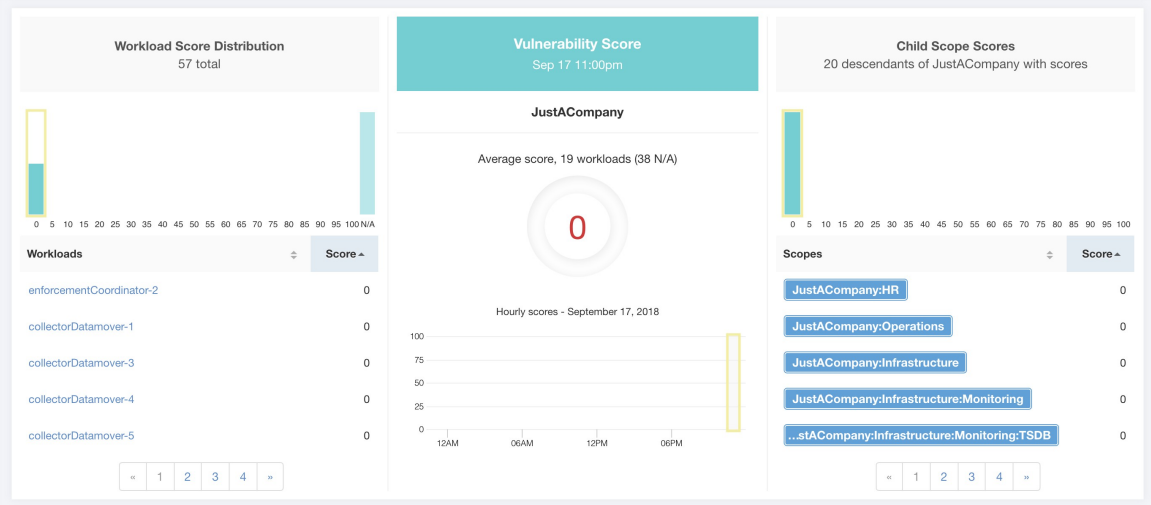


このセクションでは、各スコアカテゴリの詳細について説明します。

脆弱性セキュリティスコア

ワークロードにインストールされたソフトウェアパッケージの脆弱性は、脆弱性セキュリティスコアの計算に使用されます。

図 *530 :* 脆弱性セキュリティスコアの詳細



低いスコアは次のことを示しています。

- インストールされている 1 つ以上のソフトウェアパッケージに深刻な脆弱性がある
- パッチまたはアップグレードを適用して、セキュリティリスクやエクスプロイトの可能性を減らせる

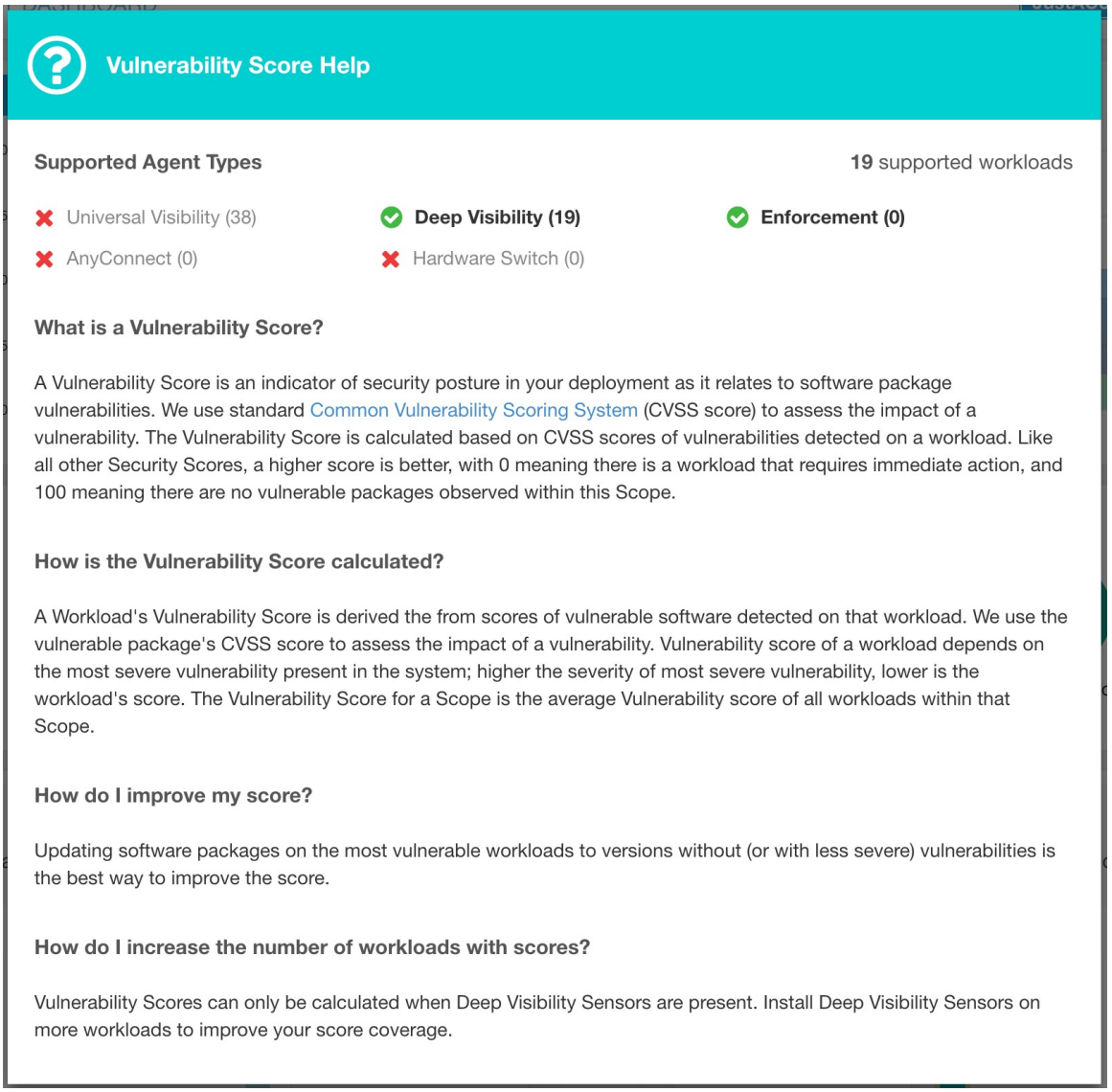
ワークロード上のソフトウェアパッケージは、既知の脆弱性（CVE）に関連付けられている可能性があります。CVSS（共通脆弱性評価システム）は、CVE の影響を評価するために使用されます。CVSS スコアの範囲は 0 ～ 10 で、10 が最も重大です。

CVE には、CVSS v2 および CVSS v3 スコアを設定できます。脆弱性スコアを計算するために、使用可能な場合は CVSS v3 が考慮され、それ以外の場合は CVSS v2 が考慮されます。

ワークロードの脆弱性スコアは、そのワークロードで検出された脆弱なソフトウェアのスコアから導き出されます。ワークロード脆弱性スコアは、CVSS スコアおよびベンダーデータに基づいて計算され、データが欠落しているか不正確な場合（新しい脆弱性に共通の問題）、シスコのセキュリティ調査チームによって調整される場合があります。このデータは、脅威フィードが設定されている場合、24 時間ごとに更新されます。最も重大な脆弱性の重大度が高いほど、スコアは低くなります。

範囲スコアは、範囲内のワークロードスコアの平均です。脆弱なソフトウェアパッケージでワークロードまたは範囲を特定し、より安全なパッケージでパッチまたはアップグレードすることでスコアを改善します。

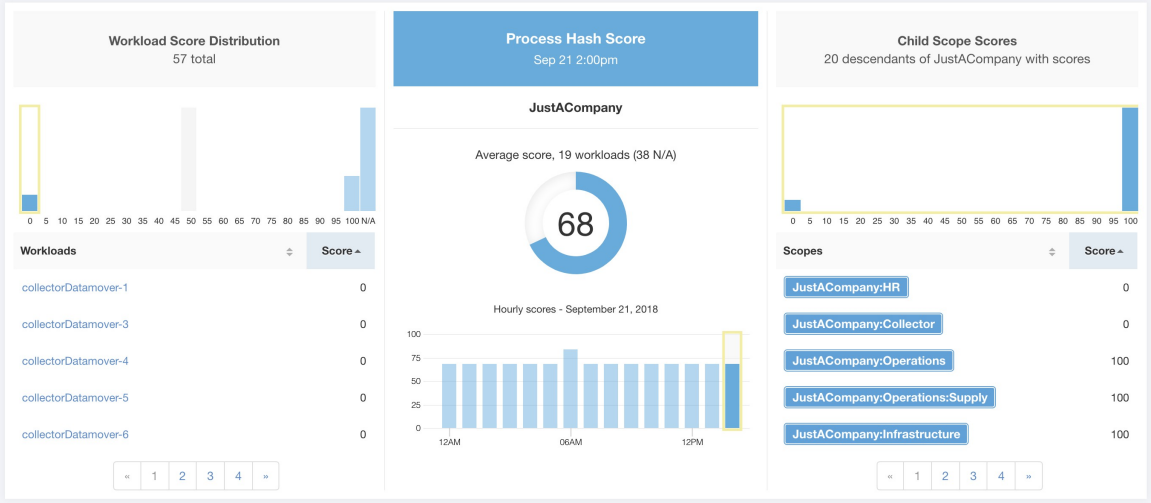
図 *531* : 脆弱性セキュリティスコアのヘルプ



プロセスハッシュスコア

プロセスハッシュスコアは、ワークロード全体でのプロセスバイナリハッシュ（ファイルハッシュ）の一貫性の評価です。例：同じセットアップ構成から複製された Apache を実行している Web サーバーファームは、すべてのサーバーの httpd バイナリに対して同じハッシュを持つことが期待されるため、不一致の場合は異常です。

図 *532 :* プロセスハッシュスコアの詳細

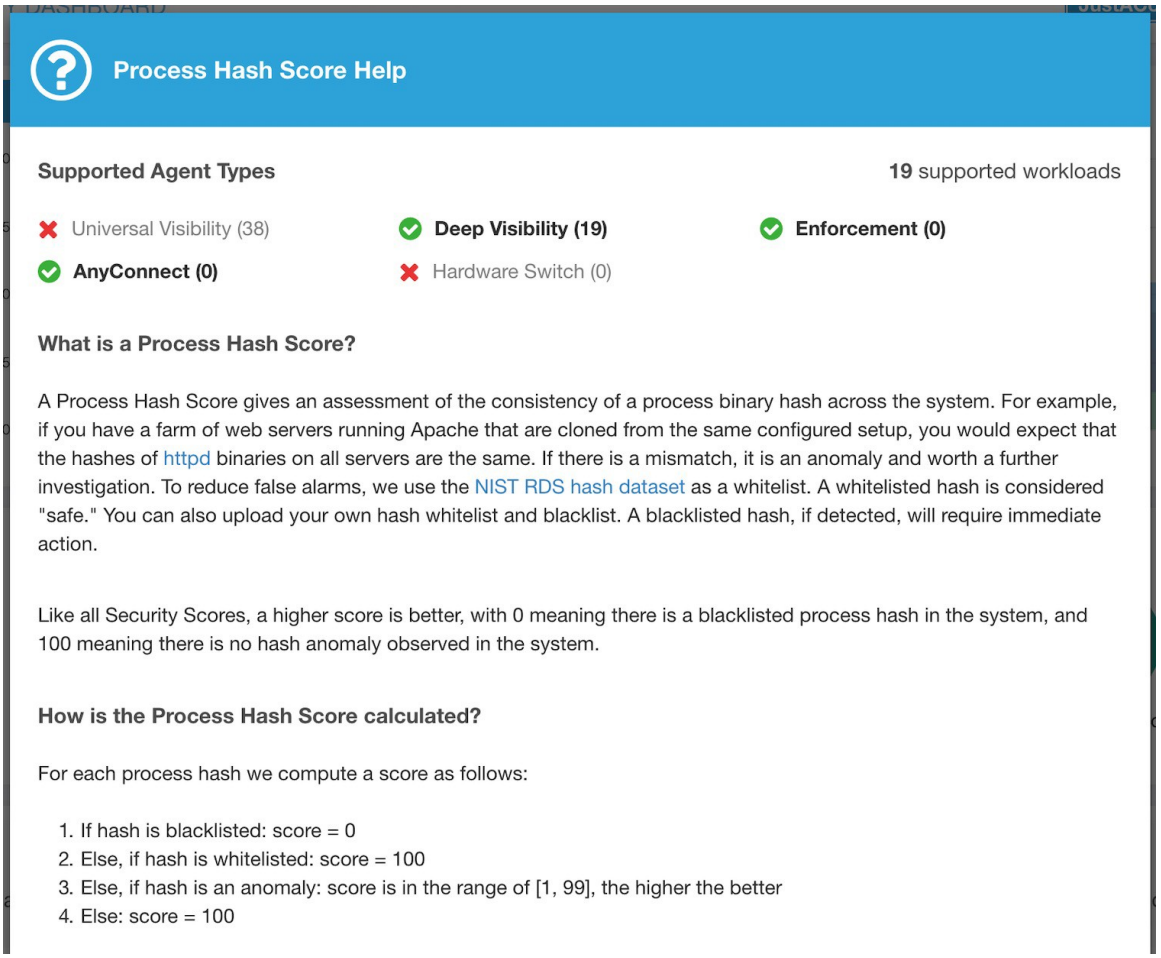


スコアが低い場合、少なくとも次のいずれか 1 つ、または両方に該当します。

- 1 つ以上のプロセスハッシュにフラグが設定されている
- 1 つ以上のプロセスハッシュが異常である

詳細については、「プロセスハッシュの異常検出」を参照してください。

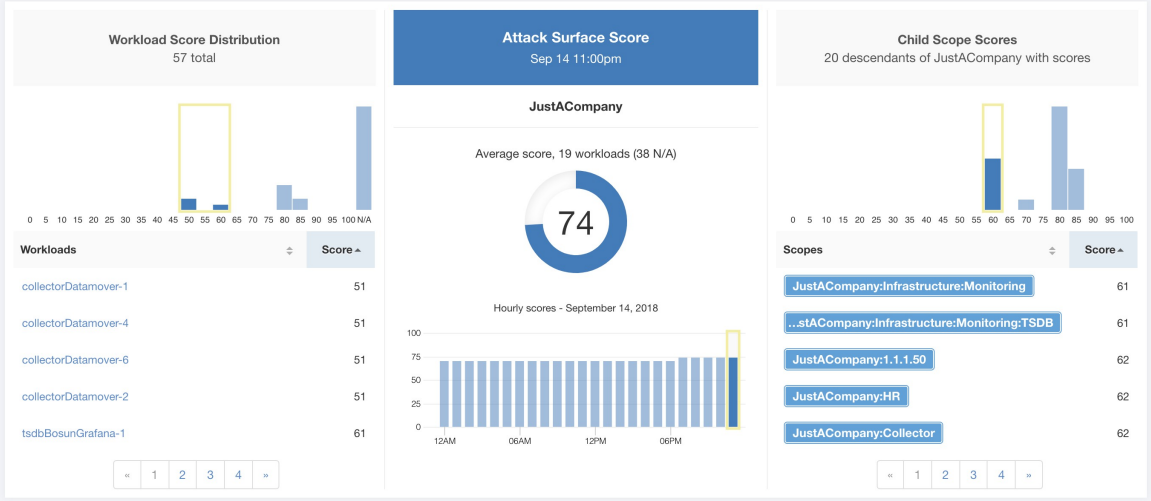
図 *533* : プロセスハッシュスコアに関するヘルプ



攻撃対象領域スコア

攻撃対象領域スコアは、ワークロードの潜在的な攻撃対象領域を強調します。未使用のオープンポート（トラフィックのないオープンポート）は、このスコアが低くなります。

図 *534* : 攻撃対象領域スコアの詳細



低いスコアは次のことを示します。

- 過去 2 週間にトラフィックのないオープンポートが多数ある
- 既知の攻撃対象ポートが、過去 2 週間オープン状態で使用されていない可能性がある
- 1 つ以上のオープンポートが、深刻な脆弱性があるパッケージに割り当てられている

攻撃対象領域スコアは、合計ポートに対する未使用オープンポートの平滑化係数を使用した関数です。過去2週間にトラフィックのないオープンポートは、「未使用のオープンポート」と見なされます。未使用のオープンポートが攻撃で使用された既知のポート（21、22、8080 など）である場合は、追加のペナルティが適用されます。

図 *535* : 攻撃対象領域スコアの計算式

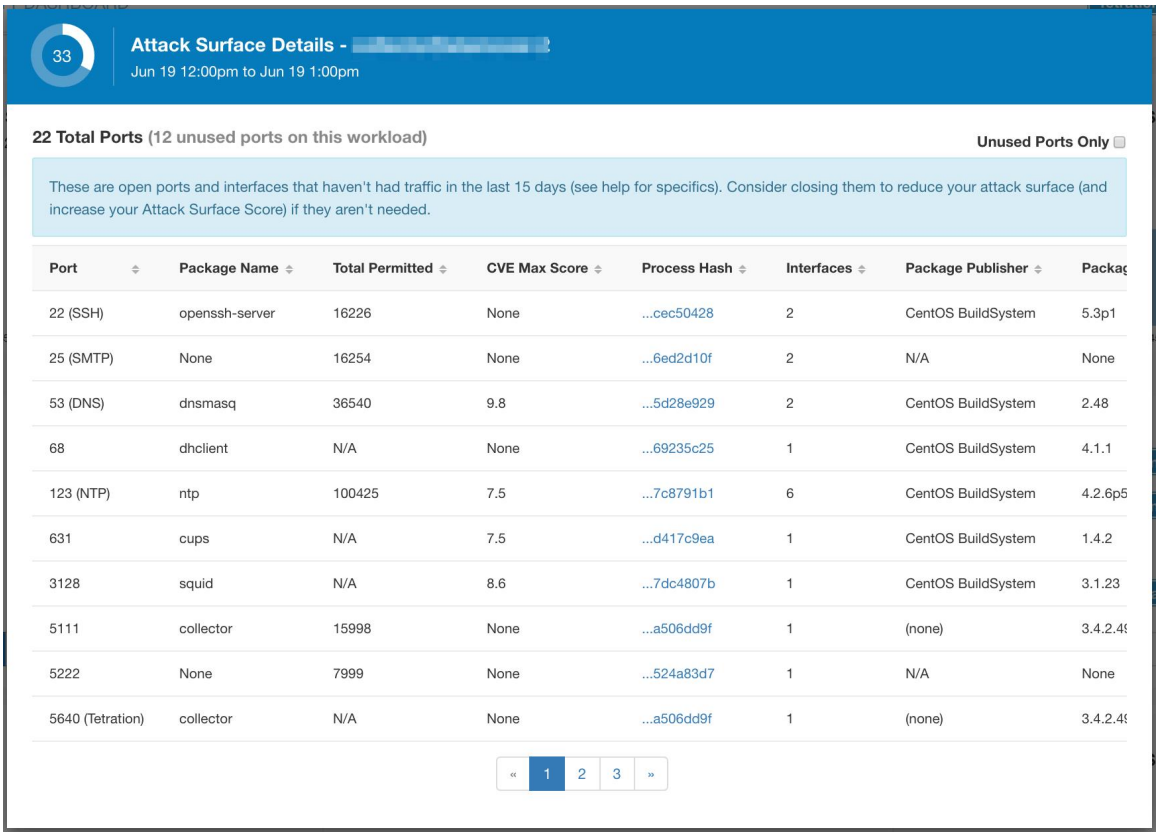
$$Attack\ surface\ score = \frac{\alpha + \sum used\ open\ ports}{\alpha + \sum open\ ports + (\rho * \sum unused\ common\ attack\ ports\ ) + f_v(vulnerablity\ pkgs)}$$

$$f_v = \max\left(\left\{cve_{score} = \begin{cases} CVSS_{V3}, & v3\ exist \\ CVSS_{V2}, & v3\ not\ exist \end{cases}\right\}\right)$$

ラプラススムージングは、ヒューリスティックデータに基づくペナルティ係数とともに使用されます。スコアは、過去 2 週間のデータを使用して毎日計算されます。

テナントスコアは、範囲内のワークロードスコアの平均です。未使用のオープンポートがあるワークロードや範囲を特定し、未使用のポートを閉じることで、スコアを改善します。

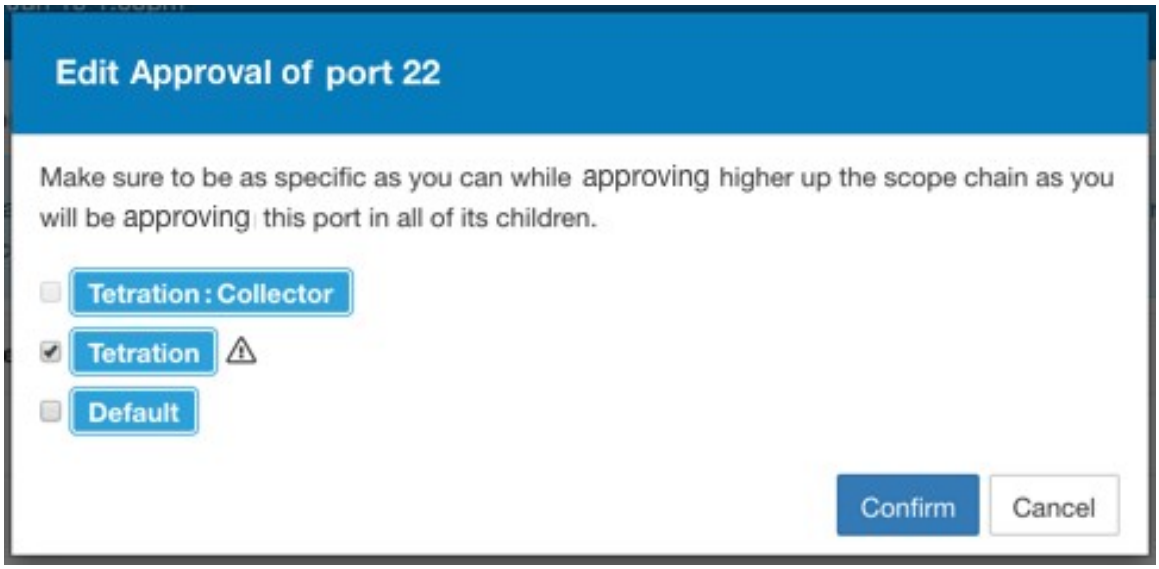
ワークロードリンクをクリックすると、攻撃対象領域モーダルが開き、そのワークロードのコンテキスト内で使用可能なすべてのポートとインターフェイスに関する詳細が表示されます。



機能：

- [未使用のポートのみ（Unused Ports Only）]：チェックボックスをオンにすると、使用中のポートが除外され、ワークロードに関連付けられている未使用ポートのみが表示されます。
- 列：[承認済み（Approved）]、[ポート（Port）]、[パッケージ名（Package Name）]、[合計許可数（Total Permitted]、[CVE最大スコア（CVE Max Score）]、[プロセスハッシュ（Process Hash）]、[インターフェイス（Interfaces）]、[パッケージ発行元（Package Publisher）]、[パッケージバージョン（Package Version）]、[合計エスケープ数（Total Escaped）]、[合計拒否数（Total Rejected）]、[一般的なハッキングポート（Commonly Hacked Port）]、[リンク（Links）]。
- インターフェイス：攻撃対象領域テーブルのいずれかの項目をクリックすると、モーダル内の各ポートに関連付けられているインターフェイスを表示できます（以下のスクリーンショットを参照してください）。
- [承認済み（Approved）]：チェックボックスをオンにすると、ワークロードがアクセスできる範囲チェーンのいずれかの範囲で、意図的に「未使用ポート」を「承認済み」として設定できます。注：ポートが範囲で承認され、そのポートがどの子範囲でも明示的に承認されていない場合（その範囲に子がある場合）、その範囲のチェックボックスは無効になります。親範囲がアクセスできる子範囲が、そのチェーンですでに承認されていることを意味するためです（以下のスクリーンショットを参照してください）。

承認済みモーダル：



インターフェイスモーダル：

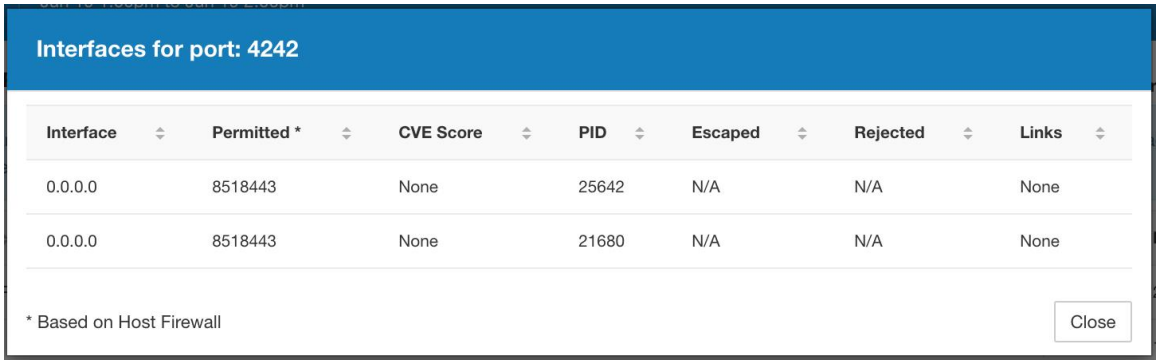
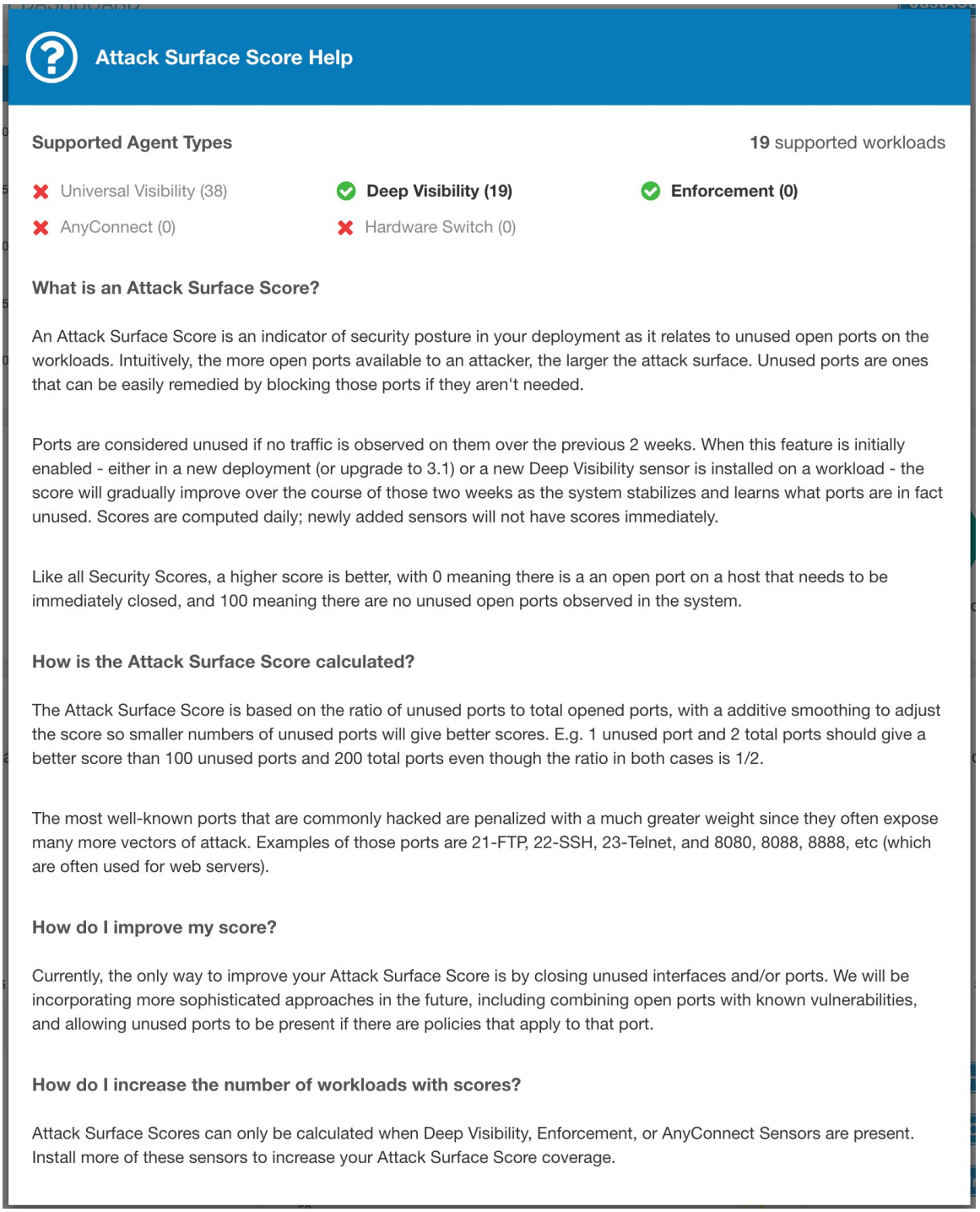


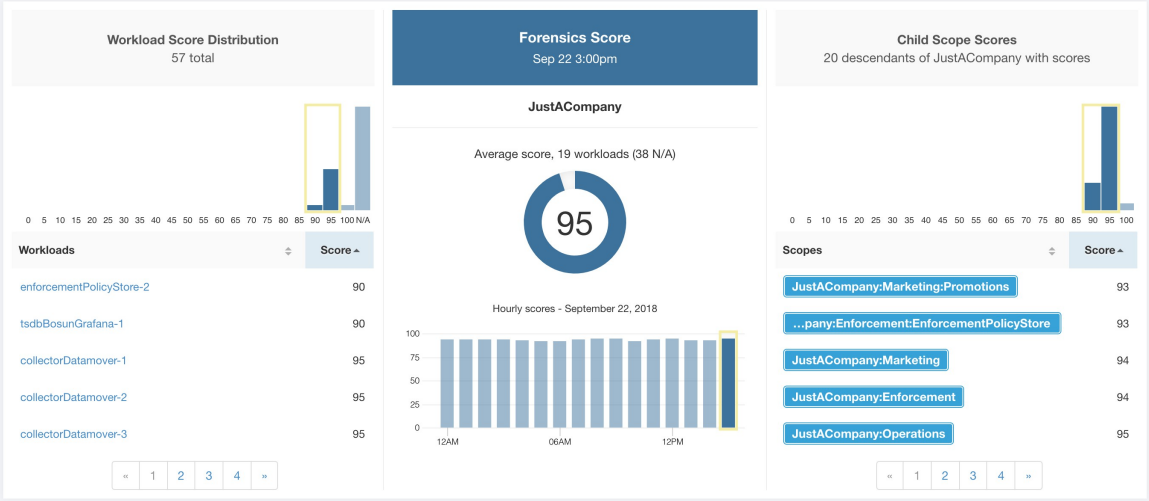
図 *536* : 攻撃対象領域スコアに関するヘルプ



フォレンジックスコア

ワークロードでのフォレンジックイベントの重大度は、このスコアの計算に使用されます。

図 *537* : フォレンジックスコアの詳細



低いスコアは次のことを示しています。

- ワークロードで 1 つ以上のフォレンジックイベントが観測された
- あるいは、1 つまたは複数のフォレンジックルールにノイズが多い、および/または正しくない

スコアを改善する方法は次のとおりです。

- 問題があれば修正して、セキュリティリスクやエクスプロイトの可能性を減らす
- フォレンジックルールを微調整して、ノイズと誤報を減らす

ワークロードのフォレンジックスコアは、フォレンジックイベントの総合影響スコアの逆関数です。フォレンジックイベントの総合影響スコアが高いほど、フォレンジックスコアは低くなります。

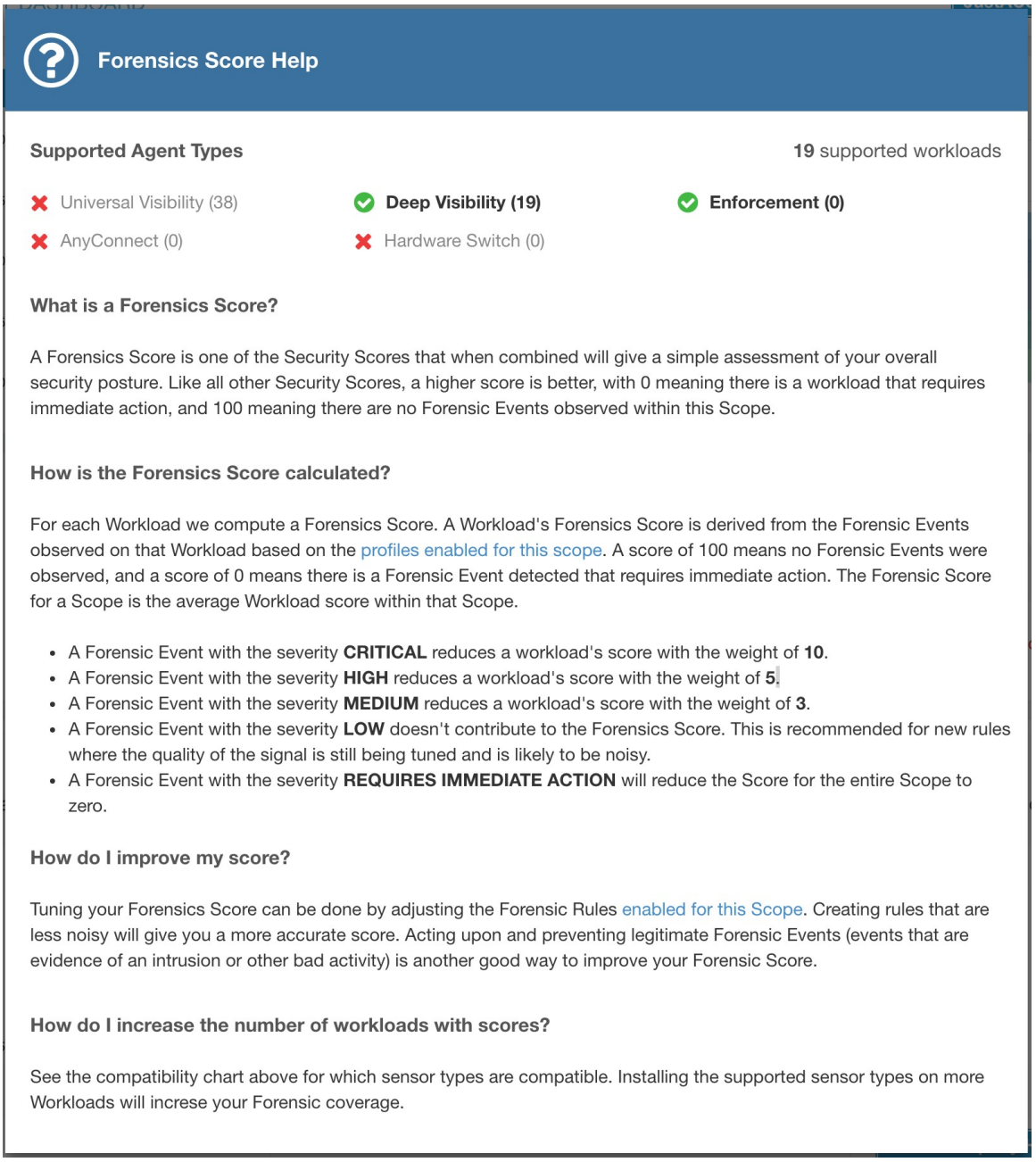
重大度	[Impact Score]
[即時対応（IMMEDIATE_ACTION）]	100
CRITICAL	10
HIGH	5
CRITICAL	3

図 *538* : フォレンジックスコアの数式

$$forensics\ score = max\left(0, (100\ -\ \sum forensics\ event\ impact\ score)\right)$$

詳細については、「フォレンジック」を参照してください。

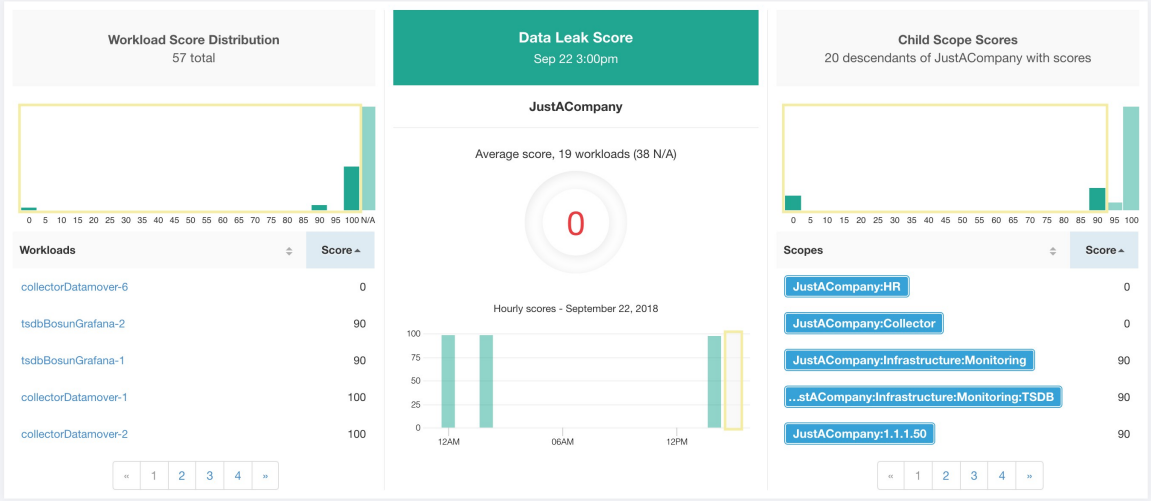
図 *539* : フォレンジックスコアのヘルプ



ネットワーク異常スコア

ワークロードのネットワーク異常イベントのシビラティ（重大度）は、スコアの計算に使用されます。

図 *540* : データ リークスコアの詳細



低いスコアは次のことを示しています。

- ワークロードから異常に大量のデータが転送されている
- またはネットワーク異常フォレンジックルールが正しくないか、ノイズが多い

スコアを改善する方法は次のとおりです。

- 問題があれば修正して、データ漏洩の可能性を減らします
- ネットワーク異常ルールを調整して、ノイズと誤報を減らします

ワークロードのネットワーク異常スコアは、ネットワーク異常イベントの合計シビラティ（重大度）スコアの逆関数です。高い方が合計シビラティ（重大度）スコアで、低い方がネットワーク異常スコアです。

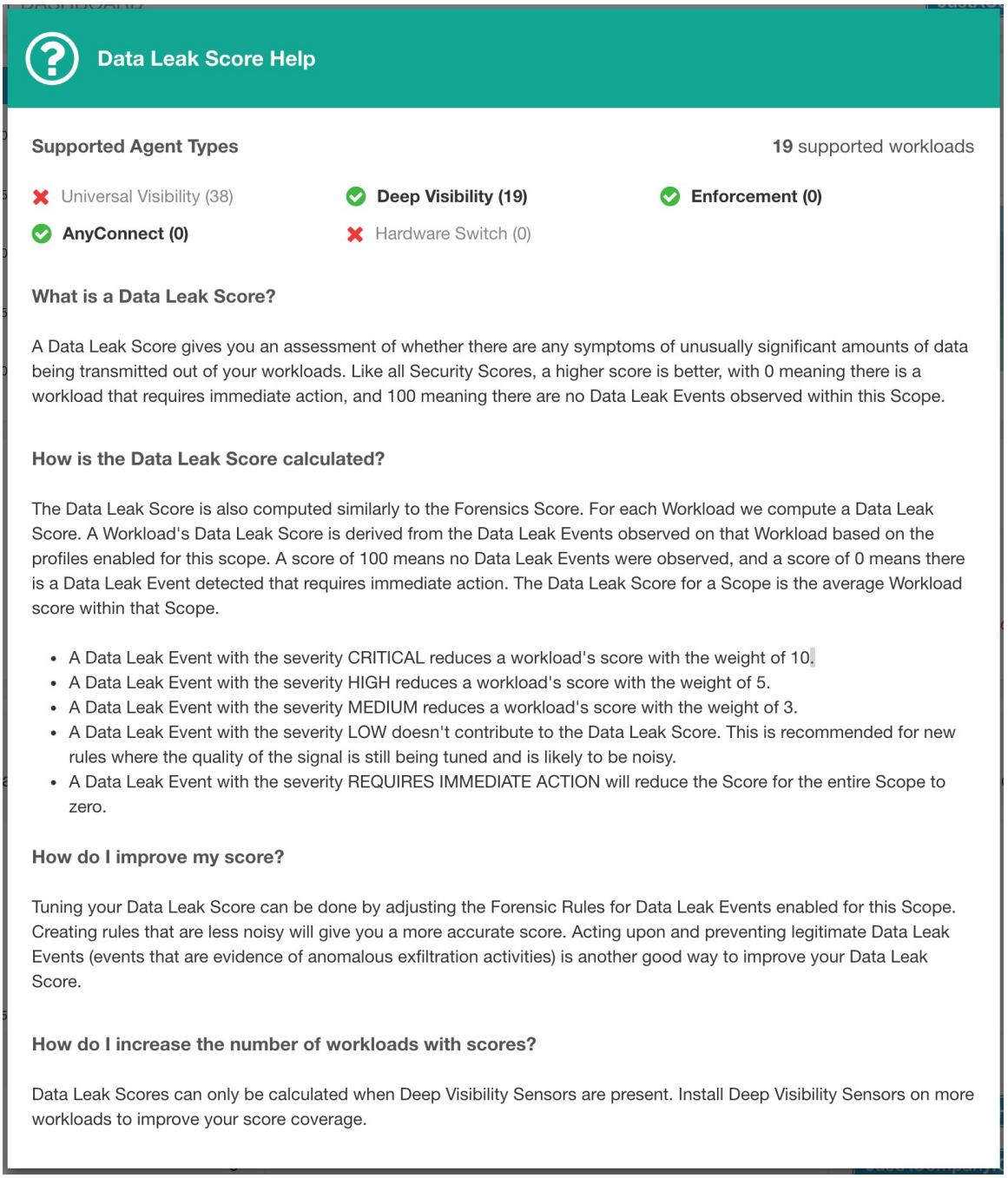
重大度	スコア
[即時対応（IMMEDIATE_ACTION）]	100
CRITICAL	10
HIGH	5
CRITICAL	3

図 *541* : データ リークスコアの数式

$$data\ leak\ score = max\left(0, (100 - \sum data\ leak\ event\ severity\ score)\right)$$

詳細については、「PCR ベースのネットワーク異常検出」を参照してください。

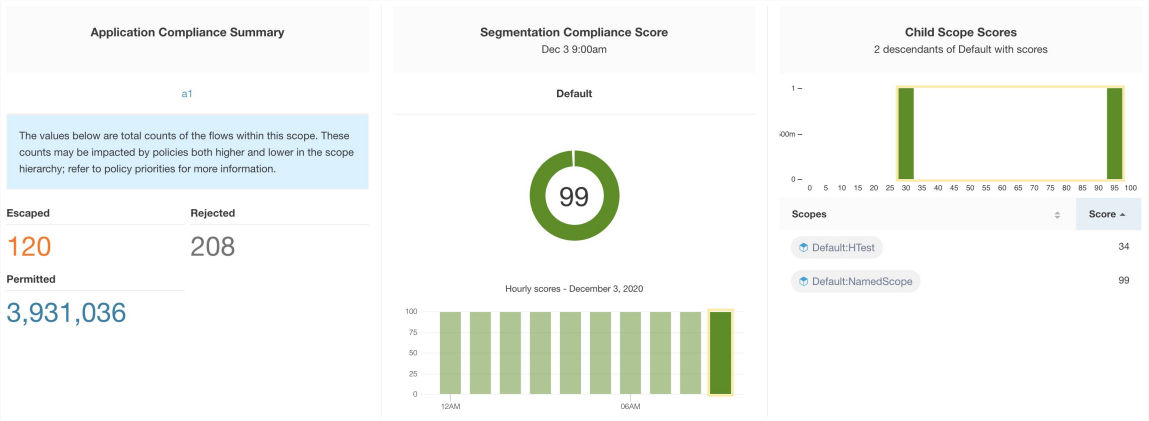
図 *542 :* データ リークスコアのヘルプ



セグメンテーション コンプライアンス スコア

セグメンテーション コンプライアンス スコアには、ポリシー違反のトップレベルのビューが表示され、最も違反が多い範囲とワークスペースが強調表示されます。

図 *543* : セグメンテーション コンプライアンス スコアの詳細



（注） ルート範囲のセキュリティダッシュボードに表示される [エスケープ/拒否/許可（Escaped/Rejected/Permitted）] カウントは、すべての子範囲についてそれぞれ表示されるすべてのカウントに加算されません。[エスケープ/拒否/許可（Escaped/Rejected/Permitted）] カウントは、送信元や宛先だけではなく、ポリシーに関する評価です。

低いスコアは次のことを示しています。

- 許可されたフローと比較して、エスケープされたフロー（ポリシー違反）の数がかなり多い
- エスケープされたフローが許可されたフローより多い場合、スコアは 0 になる

セグメンテーション コンプライアンス スコアは、プライマリワークスペースが適用されている範囲に対して計算されます。ワークスペースが適用されていない範囲の場合、スコアは、ポリシーが適用された子孫範囲スコアの平均として計算されます。

スコアは、エスケープされたフローと許可されたフローの比率を使用して計算されます。

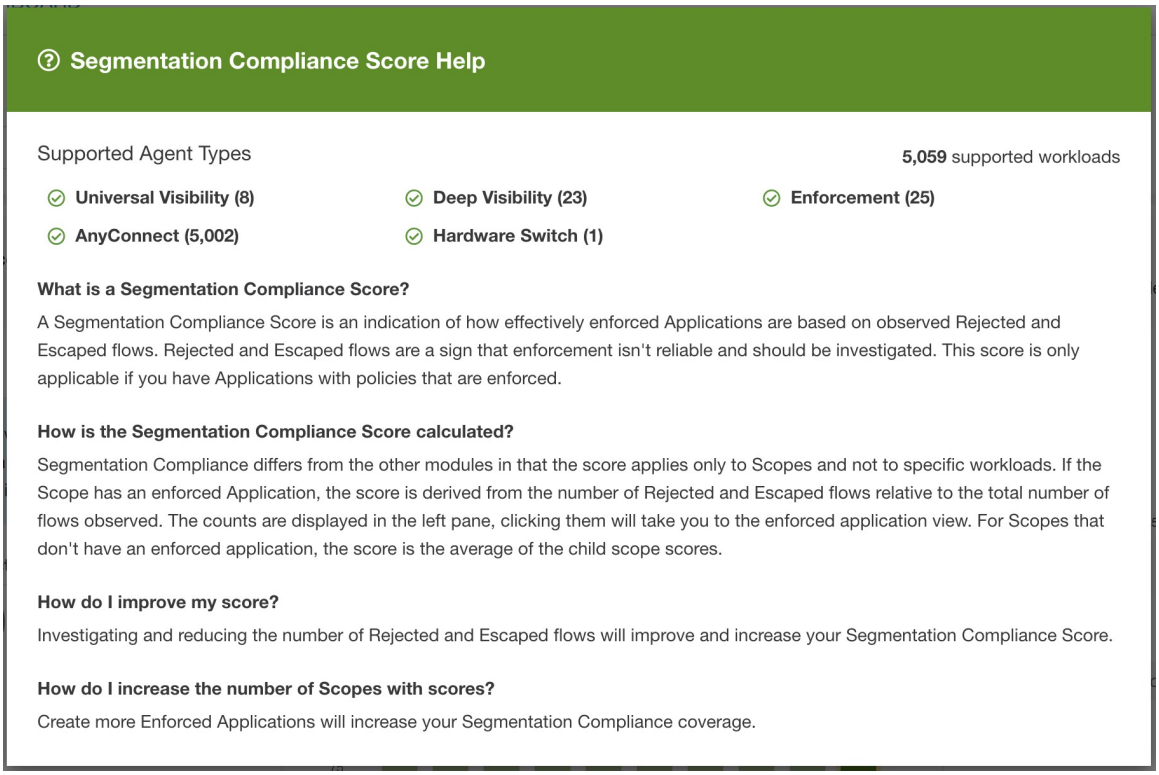
図 *544* : セグメンテーション コンプライアンス スコアの計算式

$$compliance\ score = \left\lfloor 100 - \frac{100 \times escaped}{permitted} \right\rfloor$$

ポリシー違反の数を減らしてスコアを向上させる

- ポリシーが目的の動作を正しくカバーしていることを確認する
- ポリシーが正しく適用されていることを確認する

図 *545* : セグメンテーション コンプライアンス スコアの詳細のヘルプ





第 16 章

脆弱性ダッシュボード

脆弱性ダッシュボードを使用することで、エンドユーザーは重大な脆弱性や最も注意が必要なワークロードに集中できます。ユーザーはこのページの上部で関連する範囲を選択できるだけでなく、表示する脆弱性の評価システム（共通脆弱性評価システム v2 または v3）を選択できます。新しいページには、選択した範囲の脆弱性の分布や、さまざまな属性ごとの脆弱性がハイライト表示されます。たとえば、エクスプロイトの複雑さによって、脆弱性がネットワークを介してエクスプロイトされる場合や、攻撃者がワークロードへのローカルアクセスを必要とする場合があります。さらに、リモートでエクスプロイト可能で、複雑さが最も低い脆弱性を迅速に除外するための統計情報が表示されます。

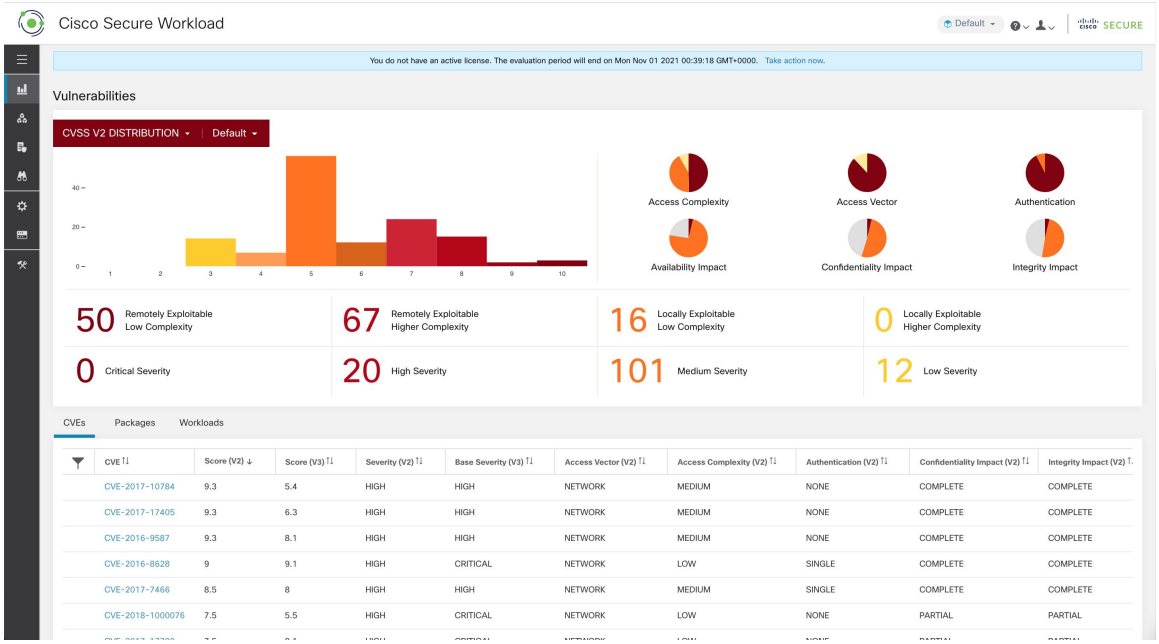
このページには 3 つのタブがあります。これらはすべて、ページ上部のウィジェットでユーザーが行ったクリックに基づいて調整およびフィルタリングされます。

- [CVE（CVEs）] タブでは、選択した範囲で注目すべき脆弱性が強調表示されます。
- [パッケージ（Packages）] タブには、パッチを適用する必要があるパッケージが表示されます。
- [ワークロード（Workloads）] タブには、選択した範囲でパッチを適用する際に最も注意が必要なワークロードが一覧表示されます。

上記のタブのいずれかの行をクリックすると、その行に関する詳細情報が表示されます。たとえば、[パッケージ（Packages）] タブのパッケージ行をクリックすると、そのパッケージ/バージョンがインストールされているワークロードと、そのパッケージに関連する脆弱性が表示されます。同様に、[ワークロード（Workloads）] タブの行をクリックすると、選択したワークロードにインストールされているパッケージと関連する脆弱性が表示されます。

このページは、最初にフォーカスするワークロードと、最初にパッチを適用するパッケージを特定するためのものです。

図 *546 :* 脆弱性ダッシュボード



- 脆弱性ダッシュボードへの移動 （970 ページ）
- [CVE] タブ （970 ページ）
- [パッケージ（Packages）] タブ （971 ページ）
- [ワークロード（Workloads）] タブ （972 ページ）

脆弱性ダッシュボードへの移動

脆弱性ダッシュボードを表示するには、ウィンドウの左側にあるナビゲーションバーで、**[調査（Investigate）]** > **[脆弱性（Vulnerabilities）]**の順にクリックします。

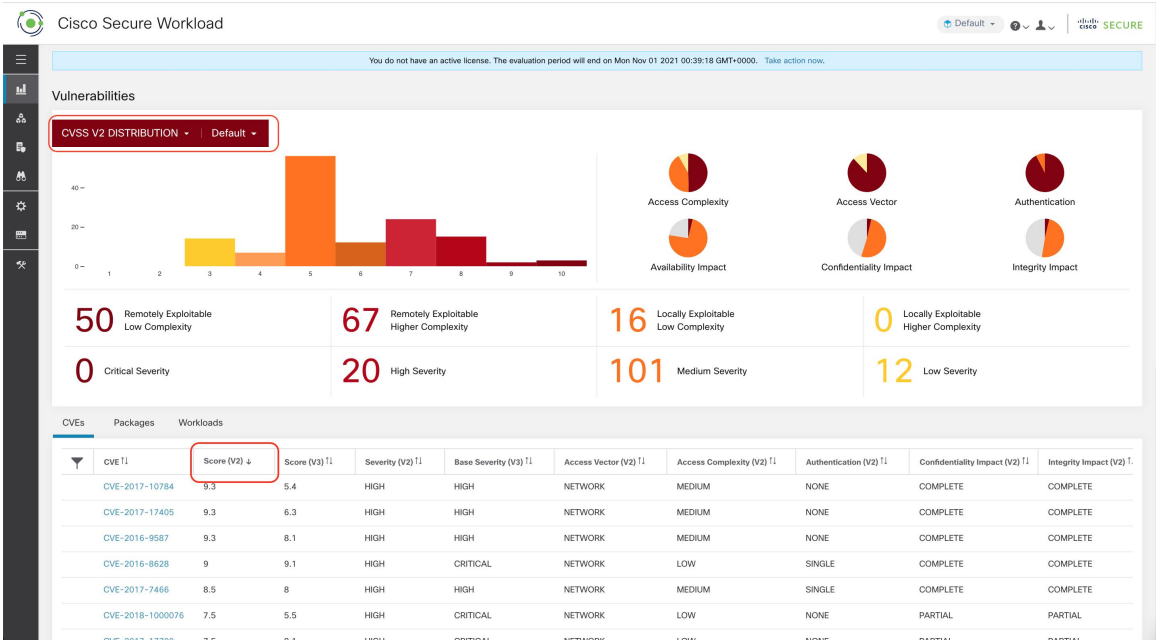
[CVE] タブ

[CVE] タブでは、ページ上部で選択した範囲とスコアリングシステム（v2 または v3）に基づいて、選択した範囲内の注意が必要なワークロードの脆弱性（スコア順に並べ替え）が強調表示されます。

CVE ごとに、基本的な影響指標に加えて、脅威インテリジェンスに基づくエクスプロイト情報が表示されます。

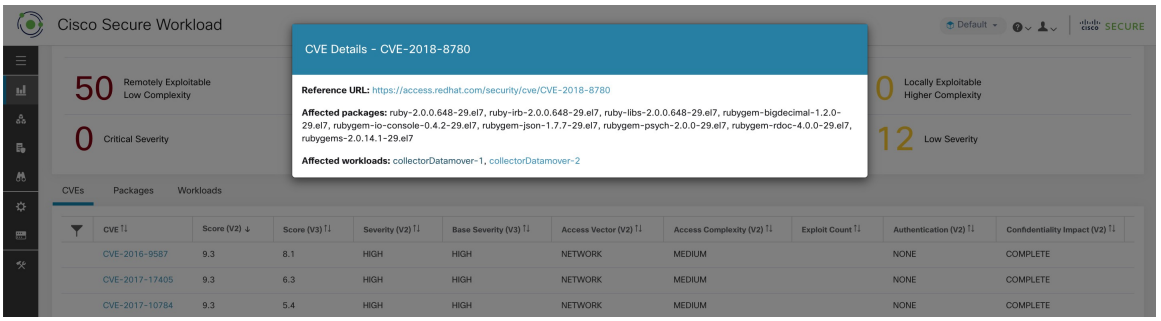
- エクスプロイト数：昨年、CVE が実際に悪用されたのが確認された回数
- 最終エクスプロイト：脅威インテリジェンスによって、CVE が実際に悪用されたのが最後に確認された時間

図 *547* : 指定された範囲の脆弱性をリストする *[CVE]* タブ



CVEテーブルのいずれかの行をクリックすると、その脆弱性と影響を受けるワークロードに関する詳細が表示されます。

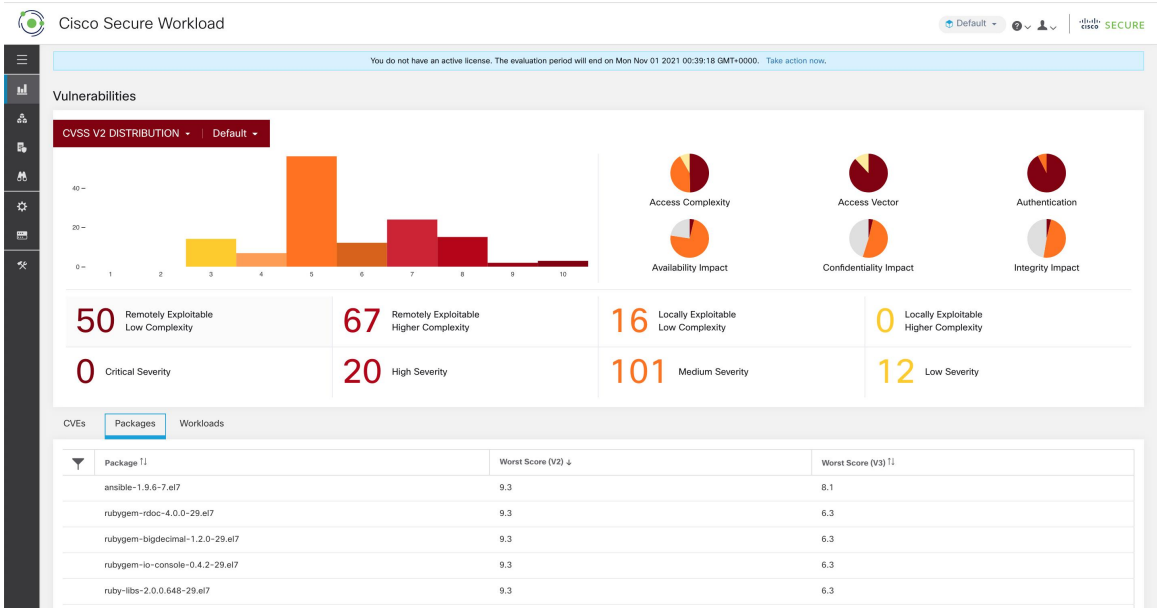
図 *548* : *CVE* の詳細



[パッケージ（Packages）] タブ

[パッケージ（Packages）] タブには、ユーザーが注意を払う必要があるソフトウェアパッケージが一覧表示されます。攻撃対象領域を減らすためにアップグレードが必要になる場合があります。

図 *549* : 特定の範囲の脆弱なソフトウェアを一覧表示する *[パッケージ（Packages）]* タブ



パッケージテーブルでいずれかの行をクリックすると、そのパッケージがインストールされているワークロードと、そのパッケージの既知の CVE に関する詳細が表示されます。

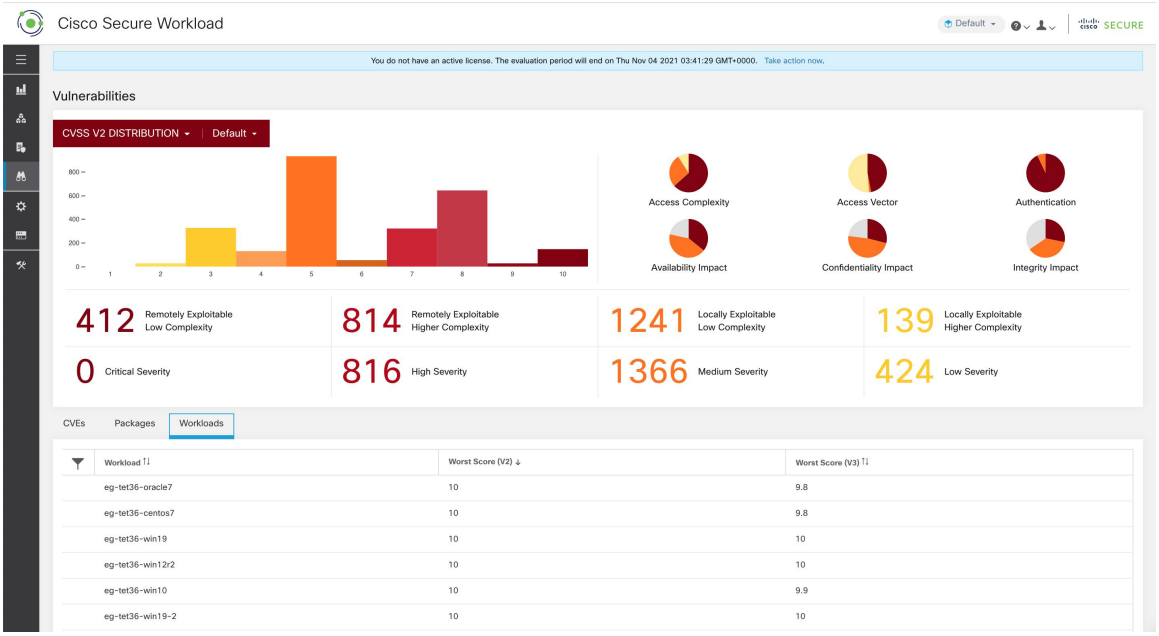
図 *550* : パッケージの脆弱性と影響を受けるワークロードの詳細



[ワークロード（Workloads）] タブ

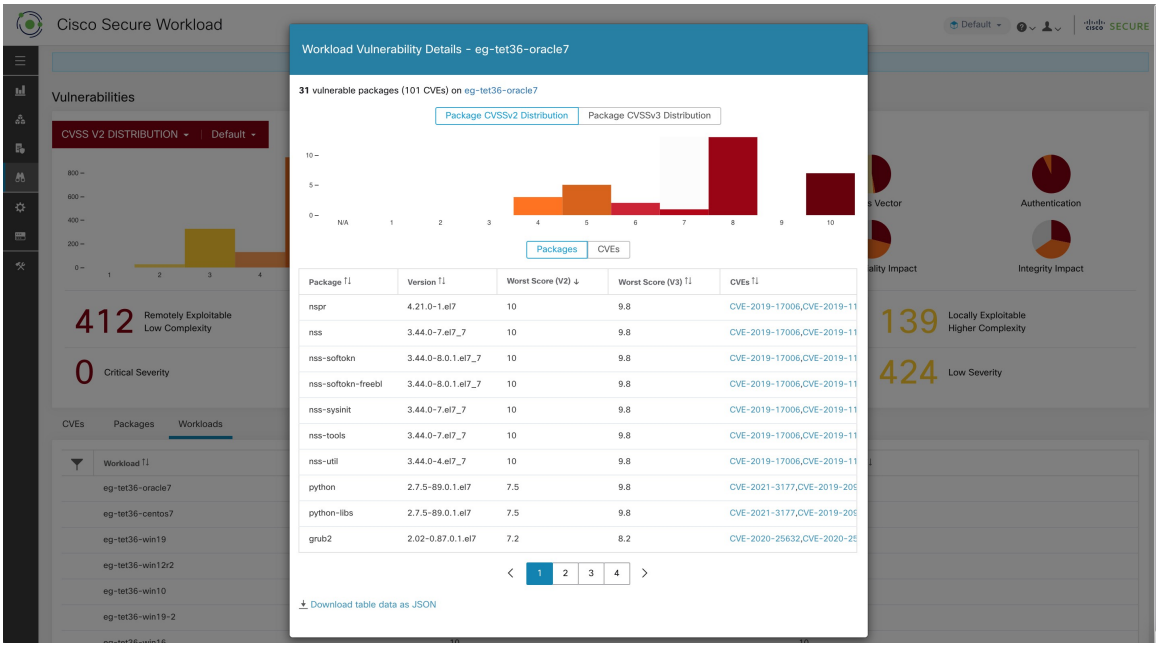
[ワークロード（Workloads）] タブには、ソフトウェアの更新またはパッチに関して注意が必要なワークロードが一覧表示されます。

図 *551* : 指定された範囲内の脆弱なワークロードが一覧表示されている *[*ワークロード（***Workloads***）*]*タブ

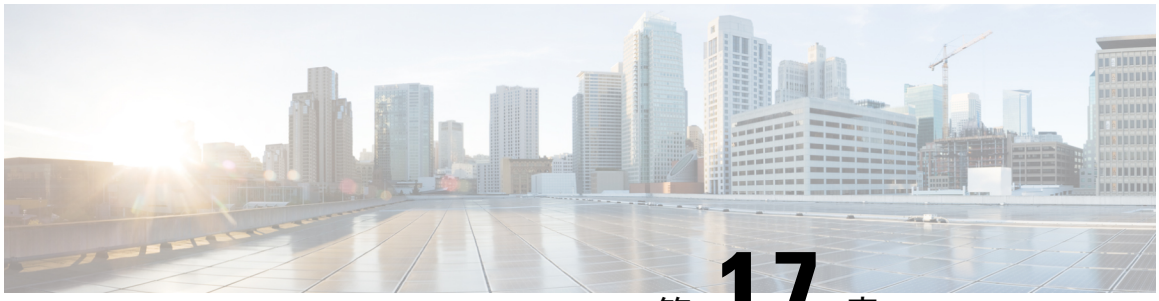


ワークロードテーブルのいずれかの行をクリックすると、そのワークロードの脆弱性があるパッケージのリストが表示されます。

図 *552* : ワークロードの脆弱性の詳細



前述のテーブルはすべて、テーブルの下部にある [ダウンロード（Download）] リンクを使用してダウンロードできます。



第 17 章

設定

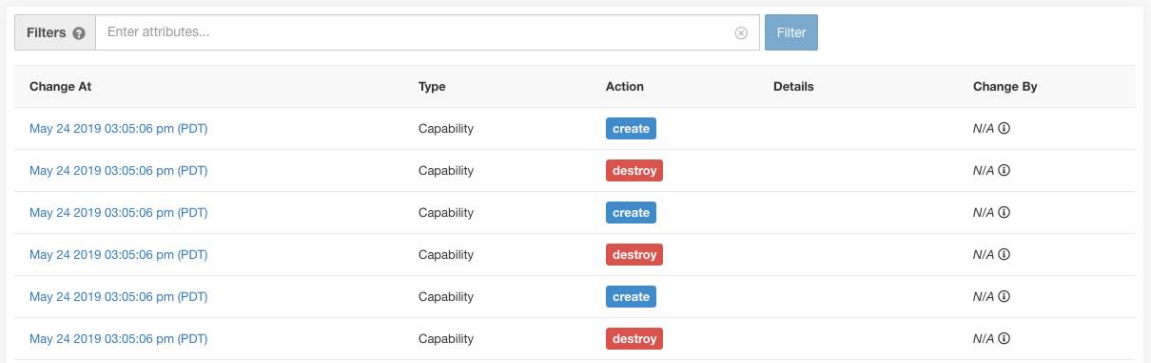
使用できるシステムレベルの設定は、ロールによって異なります。たとえば、[サイト管理者（Site Admin）] および [カスタマーサポートユーザー（Customer Support user）] ロールを持つユーザーのみが、[ユーザー（Users）] オプションを表示できます。

- ログの変更 （975 ページ）
- 収集ルール （977 ページ）
- コレクタ （978 ページ）
- 会社 （978 ページ）
- [連携（Federation）] （1001 ページ）
- アイドルセッション （1019 ページ）
- 設定 （1019 ページ）
- ロール （1023 ページ）
- スコープ （1044 ページ）
- テナント （1045 ページ）
- ユーザ（Users） （1046 ページ）

ログの変更

サイト管理者は、ウィンドウの左側にあるナビゲーションバーの [管理（Manage）] メニューの下にある [変更ログ（Change Log）] ページにアクセスできます。このページには、Cisco Secure Workload で行われた最新の変更内容がすべて表示されます。

図 *553 : [*変更ログ（*Change Log*）*]* ページ



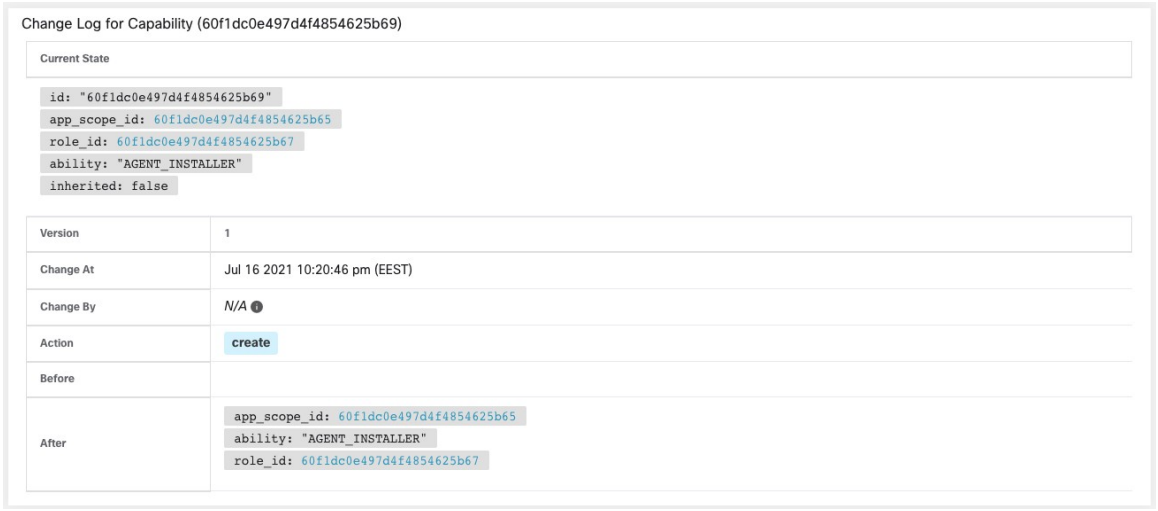
各変更ログエントリの詳細は、[変更日時（Change At）]列のリンクをクリックすると表示できます。このページには、変更されたフィールドの変更前と変更後のスナップショットが、[変更前（Before）] と [変更後（After）] に表示されます。フィールドには技術名が含まれる場合があります。Secure Workload 全体を通して見たときに、他の場所でどのような使われ方をしているのかを理解するには、何らかの解釈が必要になります。

図 *554 : [*変更ログ詳細（*Change Log Details*）*]* ページ



エンティティの変更に関する完全なリストは、右上隅にある [この<エンティティタイプ>の完全なログ（Full log for this <entity type>）] というタイトルのボタンをクリックすると表示できます。このページには、各変更の詳細が表示されます。また、エンティティの現在の状態に関する情報がある場合は、[現在の状態（Current State）] に表示されます。

図 *555 :* エンティティの完全な変更ログ



収集ルール

サイト管理者と**カスタマーサポートユーザー**は、ウィンドウの左側にあるナビゲーションバーの [管理（Manage）] メニューから [収集ルール（Collection Rules）] ページにアクセスできます。このページには、Cisco Secure Workload エージェントを実行しているスイッチで使用される VRF 別のハードウェア収集ルールがすべて表示されます。各 VRF ごとにテーブルの行があります。

ルール

VRF の [編集（Edit）] ボタンをクリックして、その収集ルールを変更します。デフォルトでは、すべての VRF は 2 つのデフォルトのキャッチオールルールによって設定されます。1 つは IPv4 (`0.0.0.0/0 INCLUDE`) 用で、もう 1 つは IPv6 (`::/0 INCLUDE`) 用です。これらのデフォルトルールは削除できますが、慎重に行ってください。

さらなる包含ルールと除外ルールを追加できます。有効なサブネットを入力し、包含または除外を選択して、[ルールの追加（Add Rule）] をクリックします。これらのルールの優先度は、ドラッグアンドドロップで調整できます。リスト内のルールをクリックしたままドラッグして、順序を調整するだけです。

変更がスイッチに反映されるまでに数分かかる場合があります。VRF リストに戻るには、右上隅の [戻る（Back）] ボタンをクリックします。

優先順位

収集ルールは、優先順位の降順に並べられます。優先順位を決定するために、最長プレフィックスの一致は行われません。最初に表示されるルールは、後続のすべてのルールよりも優先されます。例：

1. 1.1.0.0/16 INCLUDE
2. 1.0.0.0/8 EXCLUDE
3. 0.0.0.0/0 INCLUDE

上記の例では、サブネット 1.1.0.0/16 を除いて、サブネット 1.0.0.0/8 に属するすべてのアドレスが除外されています。

順序を変更した別の例：

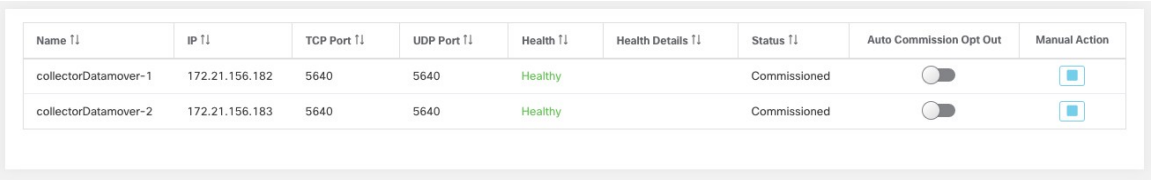
1. 1.0.0.0/8 EXCLUDE
2. 1.1.0.0/16 INCLUDE
3. 0.0.0.0/0 INCLUDE

上記の例では、サブネット1.0.0.0/8に属するすべてのアドレスが除外されています。ルール番号2は、サブネットに対してすでに高次のルールが定義されているため、ここでは実行されません。

コレクタ

サイト管理者と**カスタマーサポートのユーザー**は、ウィンドウの左側にあるナビゲーションバーの [プラットフォーム（Platform）] メニューの下にある [コレクタ（Collectors）] ページにアクセスできます。このページには、現在構成されているすべてのコレクタが表示されます。Cisco Secure Workload エージェントは、コミッションされたコレクタにフローデータを送信するため、コミッションされたすべてのコレクタが利用可能であることが重要です。デフォルトでは、すべてのコレクタは定期的に正常性をチェックされ、正常性に基づいてコミッションまたはデコミッションされます。[自動コミッションのオプトアウト（Auto Commission Opt Out）] トグルを使用して、この自動化されたプロセスからオプトアウトできます。このトグルをオンにすると、右端の列の下にある [再生（Play）] アイコンと [停止（Stop）] アイコンを使用して、コミッションとデコミッションができます。

図 *556 : [コレクタ（Collectors）] ページ*



会社

次のように、企業全体（Secure Workload クラスタごと）の構成を設定できます。

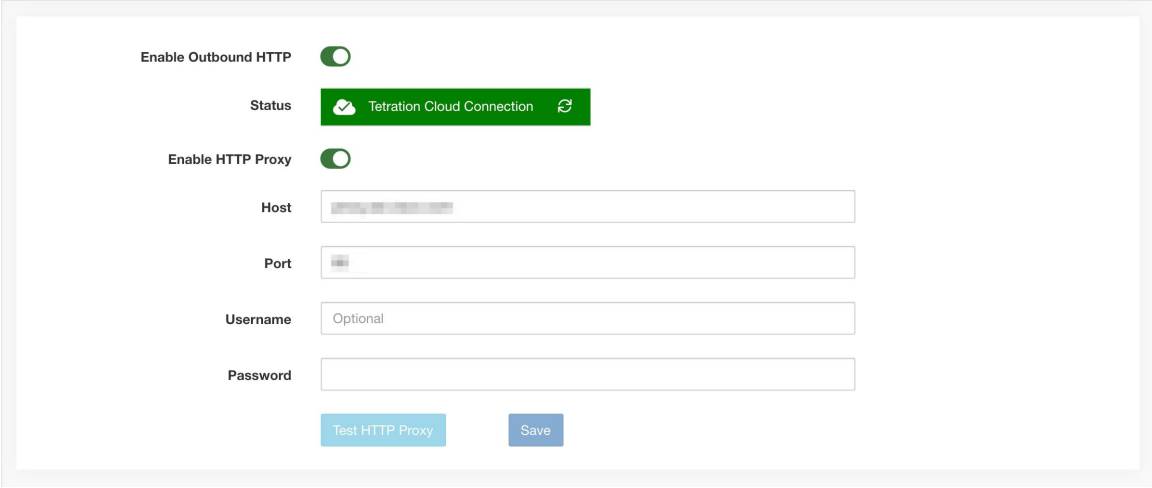
アウトバウンド HTTP 接続

Cisco Cloud から最新の脅威インテリジェンス データセットが取得されるようにするには、アウトバウンド HTTP 接続をセットアップすることを強く推奨します。

警告 エンタープライズ アウトバウンド HTTP リクエストでは、HTTP プロキシの設定に加えて、エンタープライズ ファイアウォール アウトバウンド ルールから **periscope.tetrationcloud.com** および **uas.tetrationcloud.com** へのトラフィックを許可する必要がある場合があります。以下を参照してください。

periscope.tetrationcloud.com への TLS 接続は、既知の脆弱性を識別するために脅威インテリジェンスデータを転送するために使用されます。したがって、Cisco Secure Workload では、ドメインの X.509 証明書の署名 CA 証明書を、Secure Workload に付属の信頼できるルート CA 証明書と照合して、ドメイン名の信頼性を検証することが不可欠です。X.509 信頼チェーンを改ざんすると、機能が正常に動作しなくなります。

図 *557* : アウトバウンド *HTTP* 接続



サイト管理者と**カスタマーサポートのユーザー**は、アウトバウンドHTTP設定にアクセスできます。左側のナビゲーションバーで、**[プラットフォーム（Platform）]**>**[アウトバウンドHTTP（Outbound HTTP）]**をクリックします。

フィールド	説明
[Status（ステータス）]	Secure Workload アプライアンスが Secure Workload Cloud にアクセスして脅威インテリジェンス データセットの更新を取得できるかどうかを示します。ステータス チェックは、更新ボタンをクリックして再トリガーできます。次の HTTP プロキシ設定を使用して、Secure Workload 展開に基づいて HTTP プロキシ設定を構成できます。

フィールド	説明
Enable HTTP Proxy	このオプションが有効になっている場合、すべての外部 HTTP 接続で HTTP プロキシが使用されます。
Host	HTTP プロキシホストアドレス
ポート（Port）	HTTP プロキシポート番号
Username	HTTP プロキシサーバーが Basic 認証を使用する場合にのみ必要です。
password	HTTP プロキシサーバーが Basic 認証を使用する場合にのみ必要です。

ログインページのメッセージ

サイト管理者と**カスタマーサポートユーザー**は、サインインページでユーザーに表示される最大 1600 文字のメッセージを入力できます。

ログインページメッセージを作成または変更するには:左側のナビゲーションバーで、**[プラットフォーム（Platform）]** > **[ログインページのメッセージ（Login Page Message）]** をクリックします。

セッション設定

UI ユーザー認証のアイドル セッション タイムアウトは、ここで構成できます。この構成は、アプライアンスのすべてのユーザーに適用されます。デフォルトのアイドルセッション期間は 1 時間です。アイドルセッションの継続時間は、5 分から 24 時間の範囲で設定できます。セッションタイムアウトは、この値が保存されるとすぐに、ユーザーの認証されたセッションで有効になります。

サイト管理者とカスタマーサポートユーザーは、この設定にアクセスできます。左側のナビゲーション バーで、**[管理（Manage）]** > **[セッション設定（Session Configuration）]** をクリックします。

外部認証の設定

このオプションを有効にすると、認証を外部システムに委ねることができます。認証の現在のオプションは、Lightweight Directory Access Protocol（LDAP）とシングル サインオン（SSO）です。このオプションを有効にすると、サインインするすべてのユーザーが、選択したメカニズムを使用して認証を行うようになります。特に「Use Local Authentication」オプションが有効なユーザーがいない場合は、LDAP 接続が正しく設定されていることを確認することが重要です。推奨されるアプローチは、「Use Local Authentication」オプションをオンにして、**サイト管理者**のログイン情報を持つローカル認証されたユーザーを少なくとも 1 人指定する方法で

す。このユーザーは、LDAPが正しく設定されていることを確認できます。接続が正常にセットアップされたら、ユーザー編集フローで[ローカル認証を使用（Use Local Authentication）]オプションをオフにして、このユーザーを外部認証に移行させることもできます。

サイト管理者は、外部接続の問題やユーザーサインインの失敗などのデバッグに役立つ付加的なデバッグメッセージを有効にできます。これは[外部認証のデバッグ（External Auth Debug）]オプションをオンにすることで有効にできます。これをオンにすると、付加的な説明ログメッセージが「external_auth_debug.log」という名前の別のログファイルに書き込まれます。デバッグが完了したら、[外部認証のデバッグ（External Auth Debug）]をオフにして、余分なログがログファイルに書き込まれないようにすることをお勧めします。

（注） 「Use Local Authentication」オプションに示されているように、ユーザーごとに有効にすることで、ユーザーは外部認証をバイパスできます。このオプションは、外部認証が有効になっている場合の警告メッセージを使用して、リンクからユーザー編集フローに移動する方法でも有効にすることができます。

連携が有効になっている場合は、SSOを使用した外部認証が推奨される認証アプローチです。

（注） 3.7.1.5リリース以降、外部認証セッションの削除時間が6時間から9時間に延長されました。この設定は、外部認証またはオンプレミスのみに適用されます。

サイト管理者および**カスタマーサポートユーザー**は、外部認証を設定できます。左側のナビゲーションバーで、**[プラットフォーム（Platform）]** > **[外部認証（External Authentication）]**をクリックします。

図 *558 :* 外部認証の設定

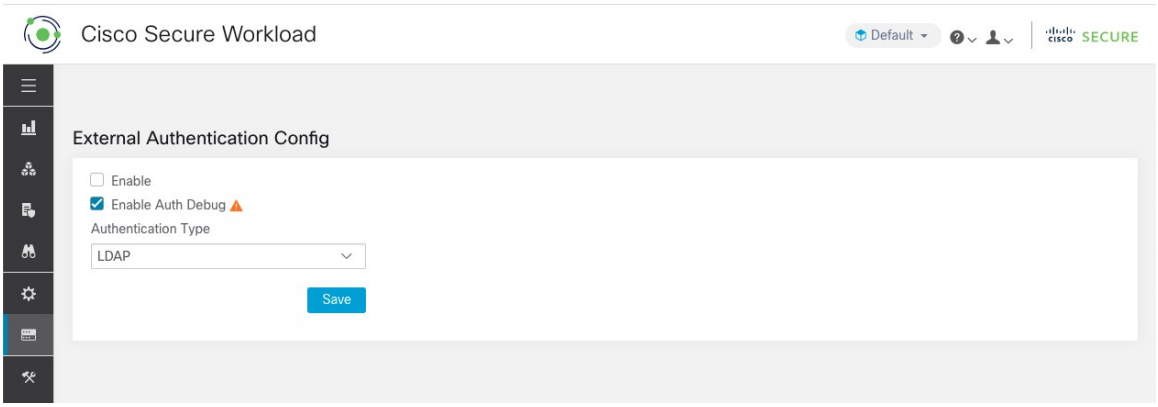


図 *559* : 外部認証の設定（続き）

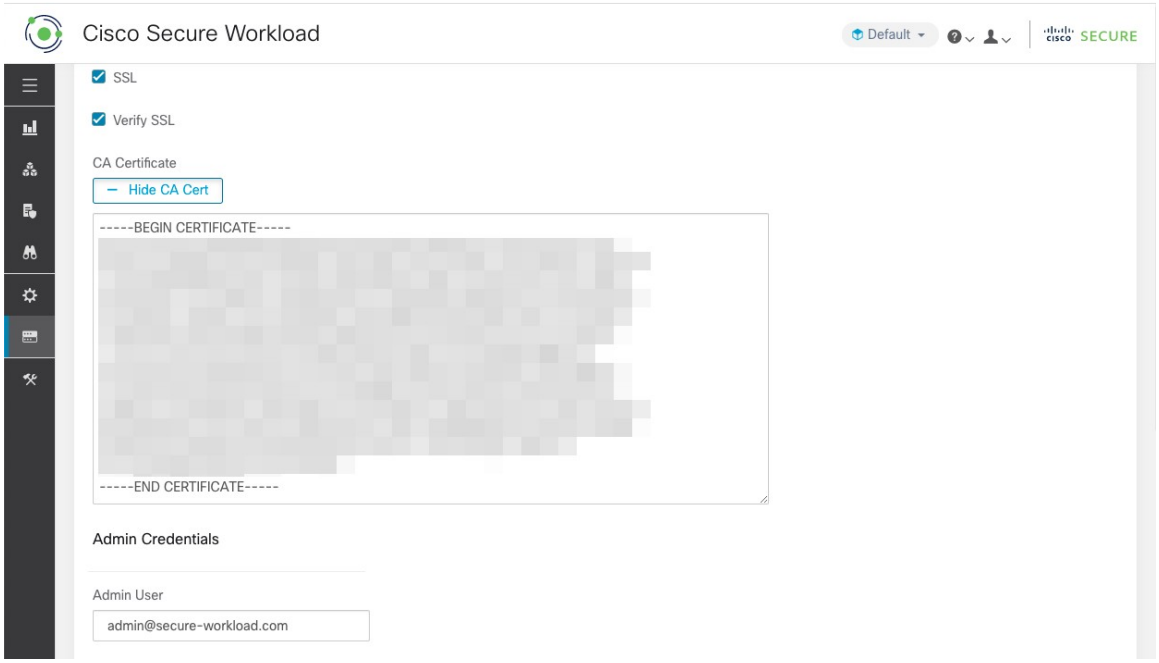


図 *560* : 外部認証の設定（続き）

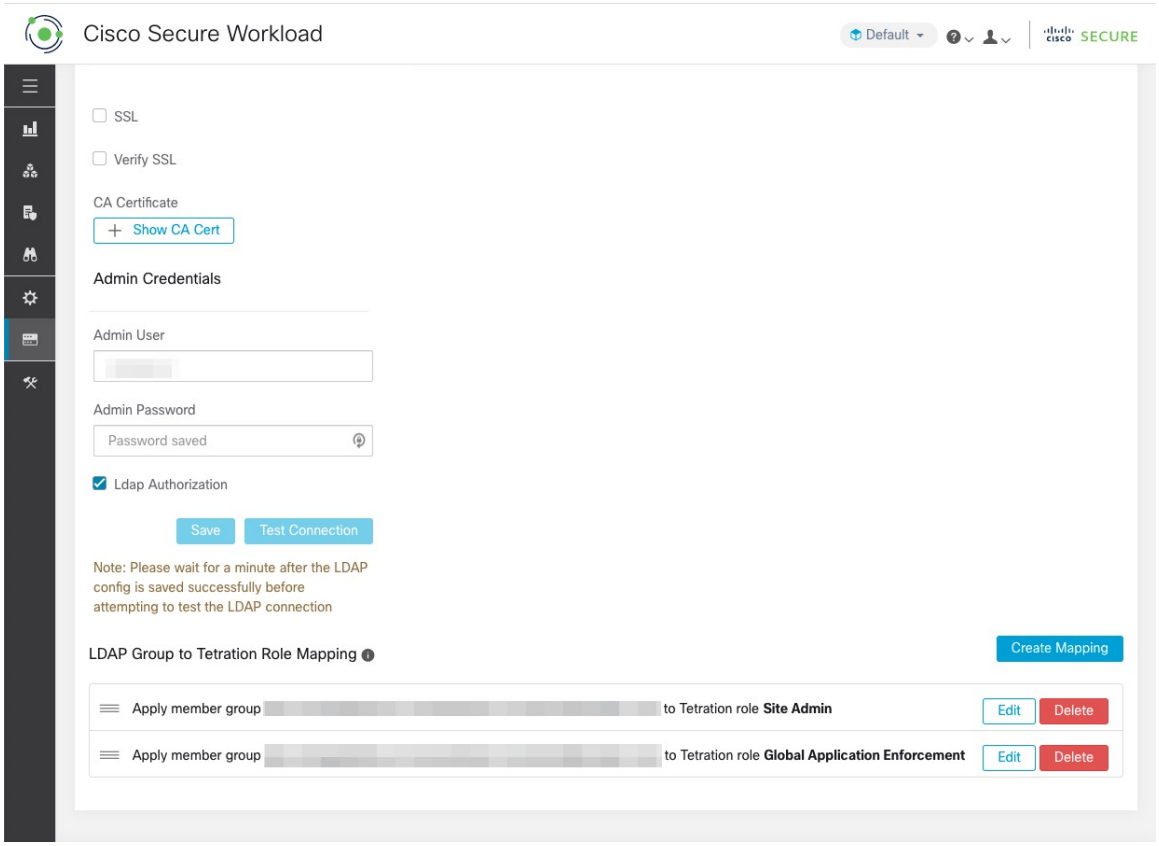
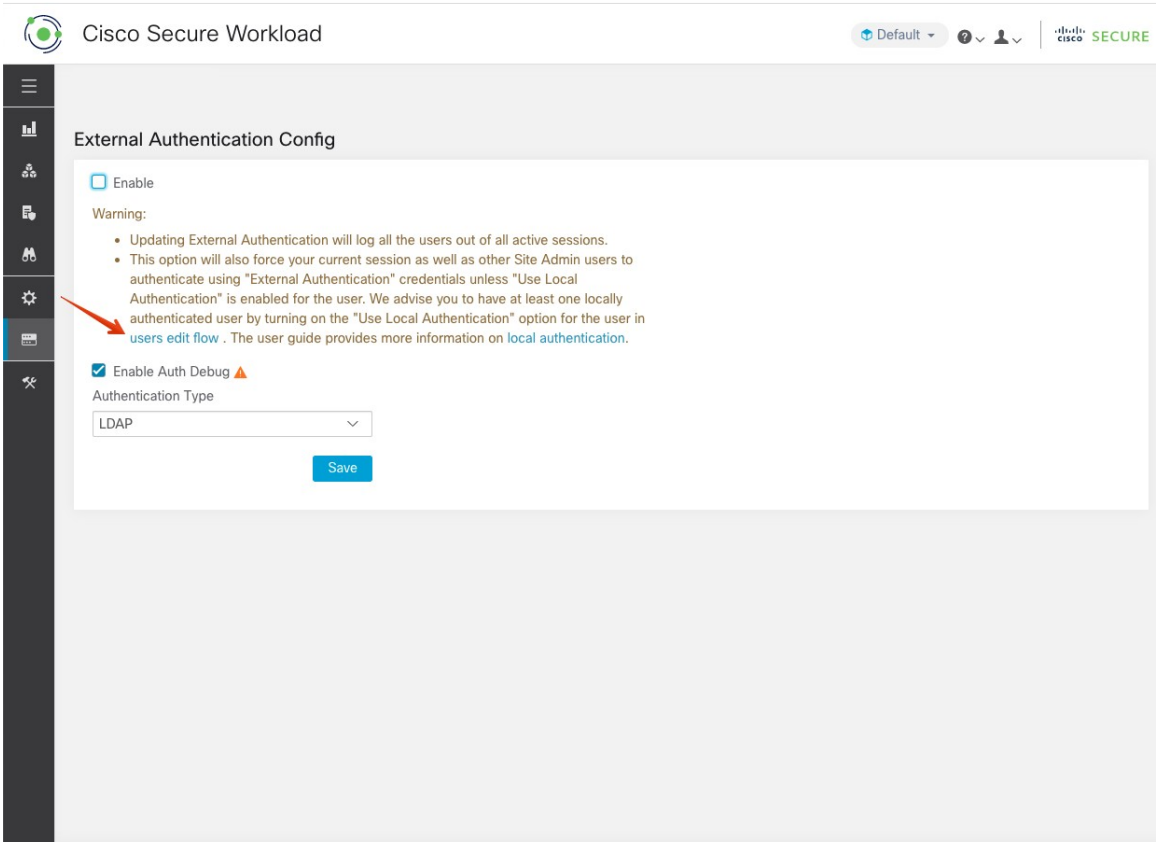


図 *561* : 外部認証に関する警告



Lightweight Directory Access Protocol の設定

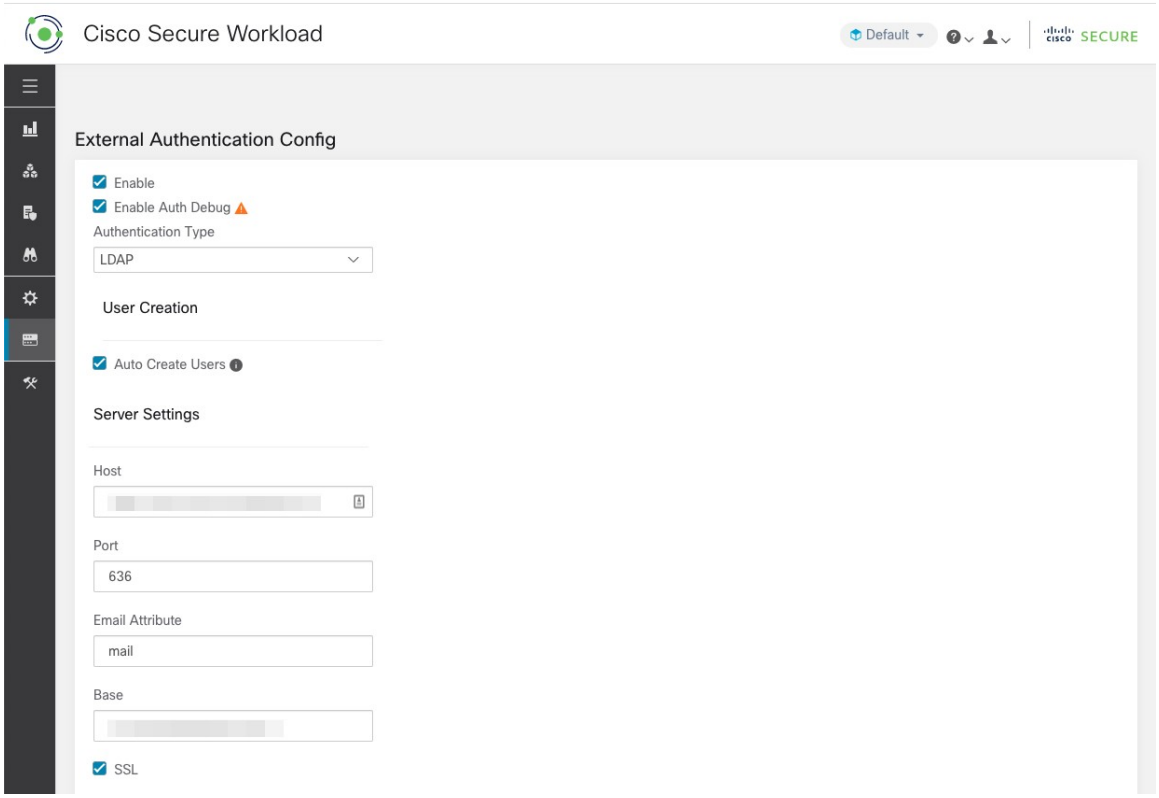
ユーザーを認証するために、Lightweight Directory Access Protocol（LDAP）オプションを選択します。つまり、これを有効にすると、すべてのユーザーがログアウトされ、その後のサインインでは LDAP の電子メールとパスワードを使用して認証されます。

「フェデレーション」が有効になっている場合、現在LDAPは認証メカニズムとして推奨されていません。

LDAPが有効になっている場合、新しいユーザーを作成するための推奨ワークフローは次のとおりです。

サイト管理者は、新しいユーザーがLDAP経由で初めてログインする前に、まず電子メールで新しいユーザーを作成し、LDAP 認証の設定（AD 認証）して適切な役割を割り当てるようにお勧めします。新しいユーザーが適切なロールなしでLDAPでログインした場合、デフォルトのロールはユーザーに割り当てられません。

図 *562 : Lightweight Directory Access Protocol* の設定



フィールド	説明
ユーザーの自動作成（Auto Create Users）	[ユーザーの自動作成（Auto Create Users）] をオンにすると、初回のログイン時に該当ユーザーが存在しない場合にユーザーが作成されます。これにより、サイト管理者は、ユーザーのログインを許可する前にユーザーを事前にプロビジョニングする必要がなくなります。Secure Workload アクセスを [ユーザー（Users）] ページで手動で作成したユーザーに制限する必要がある場合は、このオプションをオフにする必要があります。
Host	認証に使用される LDAP ホスト。
ポート（Port）	認証に使用される LDAP ポート。
メール属性（Email Attribute）	組織の電子メールを表す LDAP 属性名。
Base	ユーザーが検索される LDAP ベース DN。
SSL	暗号化を有効にして、「ldaps://」を使用します。
SSL 検証（SSL Verify）	サーバーの証明書に基づいて、完全修飾ドメイン名（FQDN）などのサーバーの SSL 属性を確認します。

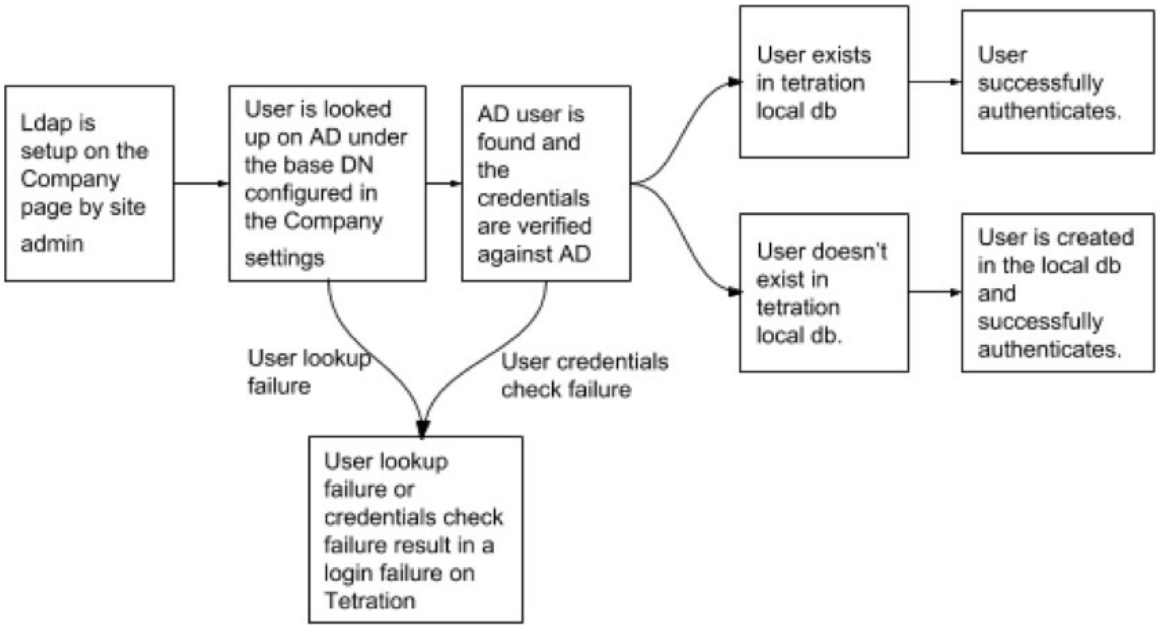
フィールド	説明
SSL認証局証明書（SSL Certificate Authority Cert）	LDAPサーバーのSSL証明書の署名証明書。サーバーの証明書チェーンを公的に検証できない場合に必要です。
管理者ユーザ（Admin User）	LDAP サーバーに対してバインドするために使用される LDAP 管理者ユーザー名（Secure Workload ユーザーではない）。例：[ユーザー]@[ドメイン] または [ドメイン]\\[ユーザー]
管理者パスワード（Admin Password）	LDAP サーバーに対してバインドするために使用される LDAP 管理者パスワード。
LDAP認証（Ldap Authorization）	LDAP認証は、「LDAP認証の設定（AD認証）」で説明されているように、有効にして構成することができます。

LDAP 構成が有効になると、「Use Local Authentication」オプションが有効になっているユーザーを除くすべてのユーザーがセッションからログアウトされます。

[保存（保存）] ボタンをクリックすると、LDAP 構成を保存できます。LDAP 構成が正常に保存された後、LDAP 接続をテストする前に 1 分間待つことをお勧めします。

[接続のテスト（Test Connection）] ボタンを使用して LDAP 構成を保存した後、LDAP 接続をテストできます。これにより、入力された管理者ログイン情報を使用してLDAPサーバーに対するバインドが試行されます。

図 *563 :* 認証ワークフロー



LDAP 問題のデバッグ

LDAP 接続のテスト時にエラーが発生した場合は、次の点を確認してください。

- LDAP 管理者の資格情報が正しいかどうかを確認します。
- ホスト、ポート、SSL などの接続パラメータを確認します。
- Secure Workload UI VIP から LDAP サーバーに到達できるかどうかを確認します。
- AD サーバーが稼働しているかどうかを確認します。
- [ldapsearch] などのコマンドラインツールを接続の詳細とともに使用して、バインドできるかどうかを確認します。

ユーザーのログイン中にエラーが発生した場合は、以下を確認してください。

- ユーザーが LDAP 認証を使用する他社の Web サイトに LDAP ログイン情報でログインできるかどうかを確認します。
- 企業の LDAP 設定で指定されている「基本の」DN が正しいかどうかを確認します。[ldapsearch] などのコマンドラインツールを使用して、基本DNに対してユーザーを検索することで実行できます。

電子メールでユーザーを検索する [ldapsearch] クエリの例：

```
ldapsearch -H "ldap://<host>:<port>" -b "<base-dn>" -D "<ldap-admin-user>" -w
<ldap->admin-password> "(mail=<users-email-address>)"
```

LDAP 認証の設定（AD 認証）

Active Directory 認証は、外部認証 LDAP 設定の [管理者資格情報（Admin Credentials）] セクションで [LDAP 認証（LDAP Authorization）] チェックボックスを有効にすることで設定できます。この設定を有効にすると、サイト管理者は、LDAP の「MemberOf」グループのマッピングを以下のセクションの Secure Workload ロールに設定する必要があります。デフォルトではこの設定がないため、Active Directory ユーザーはログインを試行する前に、1 つ以上の Secure Workload ロールを事前に設定する必要があります。

LDAP 外部認証が有効になっている場合、LDAP MemberOf グループの Secure Workload ロールへのマッピングを設定する必要があります。[マッピングの作成（Create Mapping）] を使用すると、LDAP MemberOf グループ値を Secure Workload ロールにマッピングするように設定できます。ロールドロップダウンのロールは、範囲セレクタで選択された範囲に基づいて事前に入力されています。マッピングが保存されると、すべてのユーザーは、その後のログイン時にこれらの値に基づいて承認されます。

マッピングは、並べ替え、編集、または削除ができます。マッピングへの変更は、その後のログイン時にユーザーに割り当てられたロールに反映されます。最大 50 の LDAP MemberOf グループから Secure Workload ロールへのマッピングを作成できます。

LDAP MemberOf グループ名の重複は許可されません。ただし、複数の LDAP MemberOf グループを同じロールにマッピングできます。複数のグループが同じロールにマッピングされている場合、最後のマッピングは、Secure Workload ロールに一致する LDAP MemberOf としてユーザーで保存されます。

図 *564 : Secure Workload* ロールのセットアップへの *LDAP* グループ

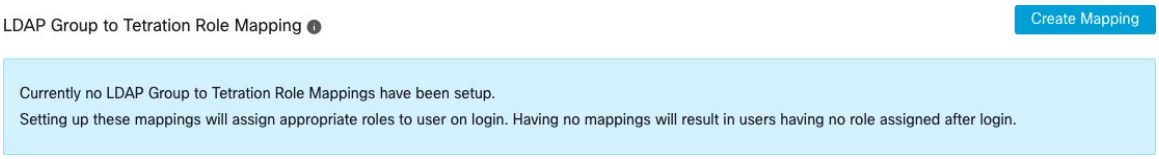


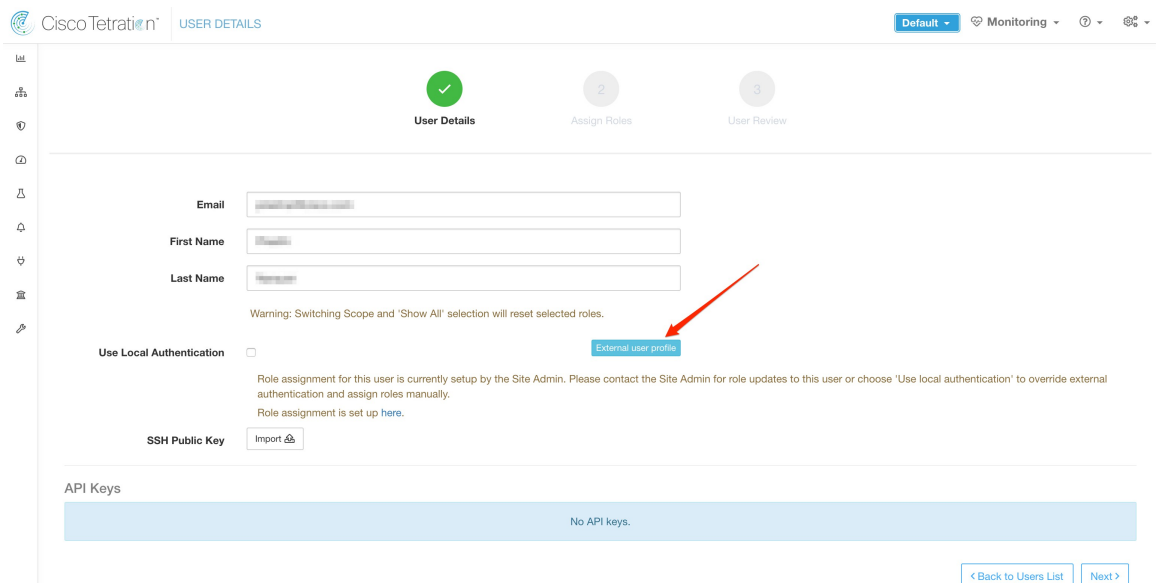
図 *565 : Secure Workload* ロールのマッピングへの *LDAP* グループ



サイト管理者ユーザーは、ユーザーが最後に成功したログインから取得した外部ユーザーの情報を利用して、上記のロールマッピングに基づいてロールの割り当てを調整できます。

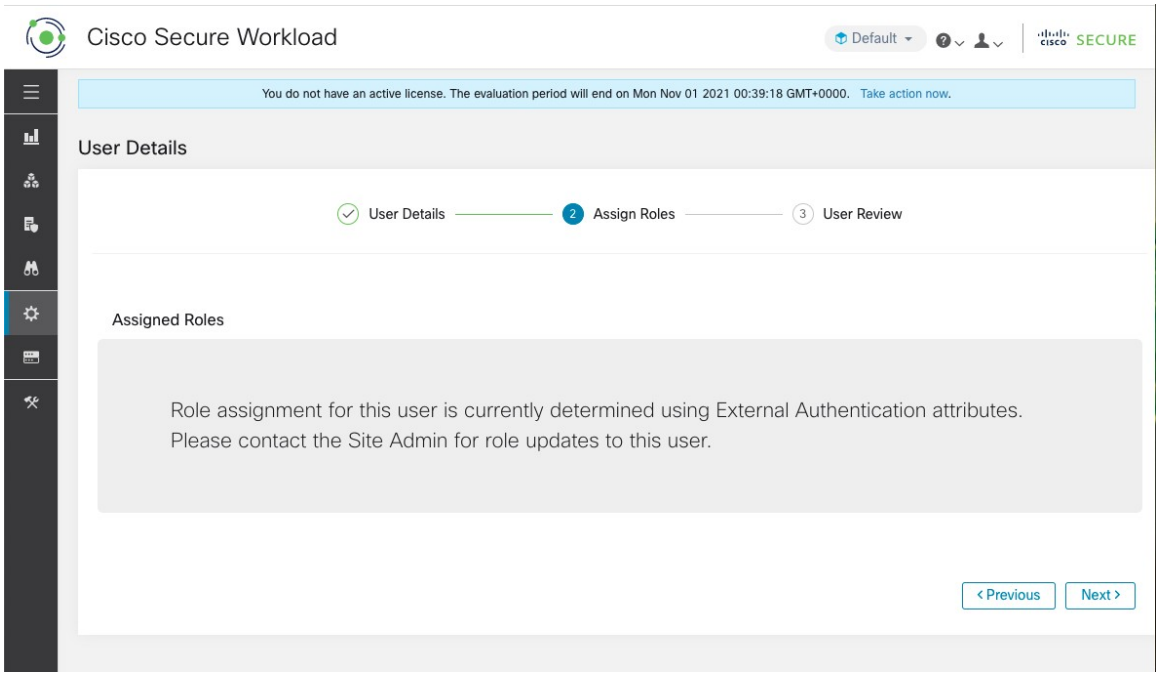
（注） 「「Use Local Authentication」オプション」オプションで示されているように、ユーザーごとに有効化すると、ユーザーは外部認証をバイパスできます。これらのユーザーは、AD 認証用に設定された認証プロセスもバイパスします。

図 *566 :* 外部ユーザー情報



認証が有効化されると、ユーザー作成フロー（新しいユーザーアカウントの追加 （1047 ページ））およびユーザー編集フロー（「ユーザー アカウントの編集」）では、手動での Secure Workload ロール選択は**許可されません**。

図 *567 : [*ユーザ（***Users***）*]* ページ



Secure Workload ロールにマッピングされた LDAP MemberOf グループは、[ユーザープロファイル（User Profile）] ページに表示されます。

図 *568 : [*ユーザープロファイル（***User Profile***）*]*ページ

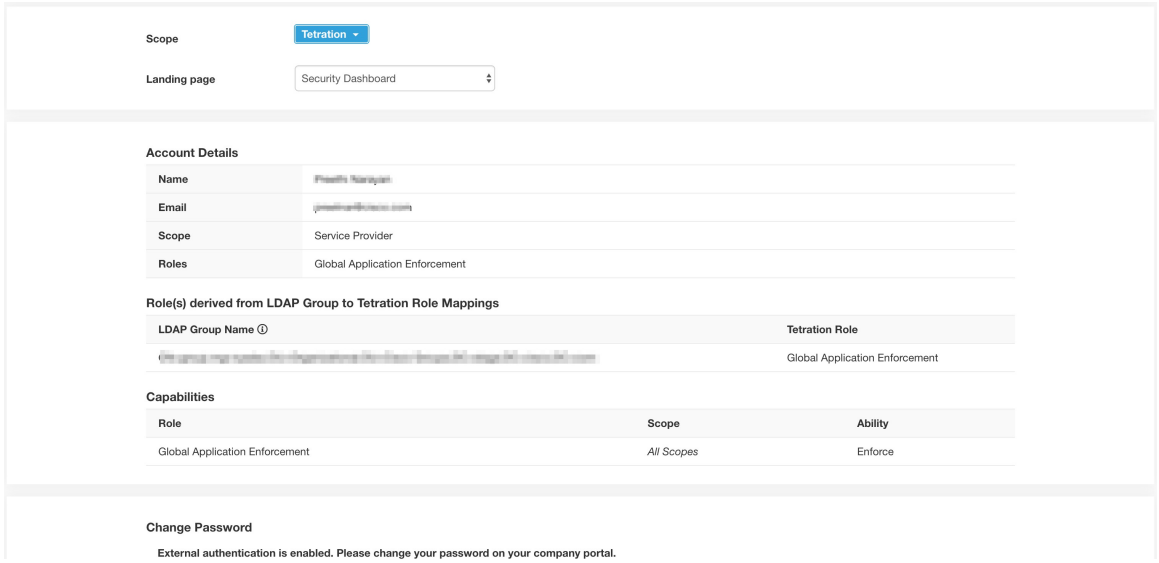
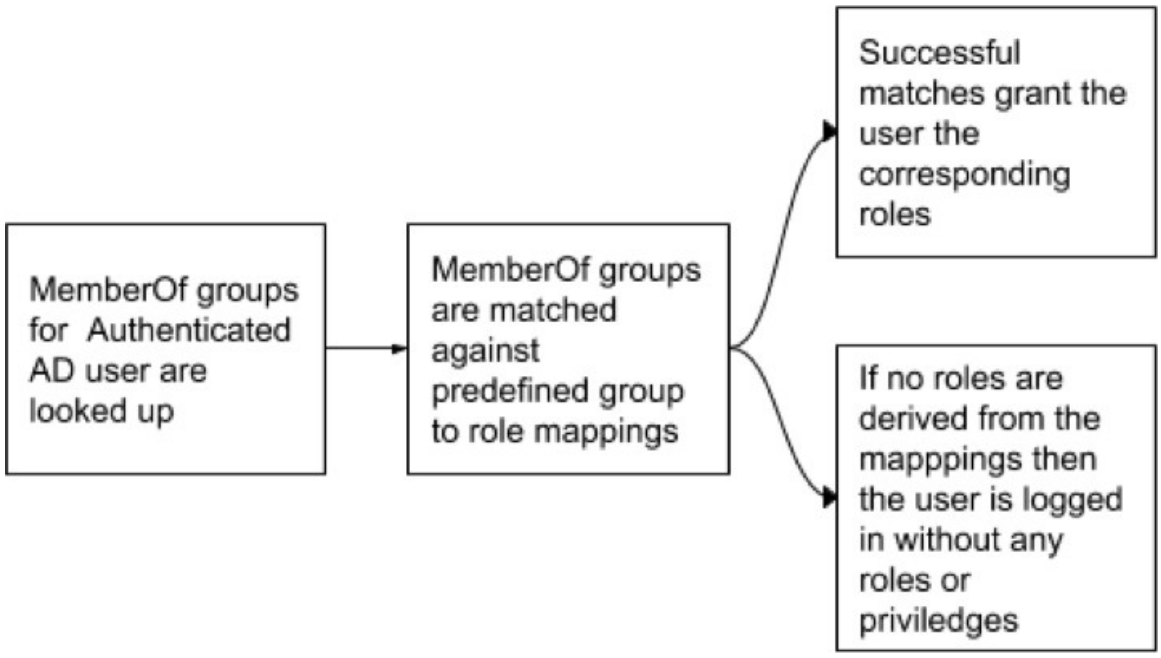


図 *569* : 認証ワークフロー

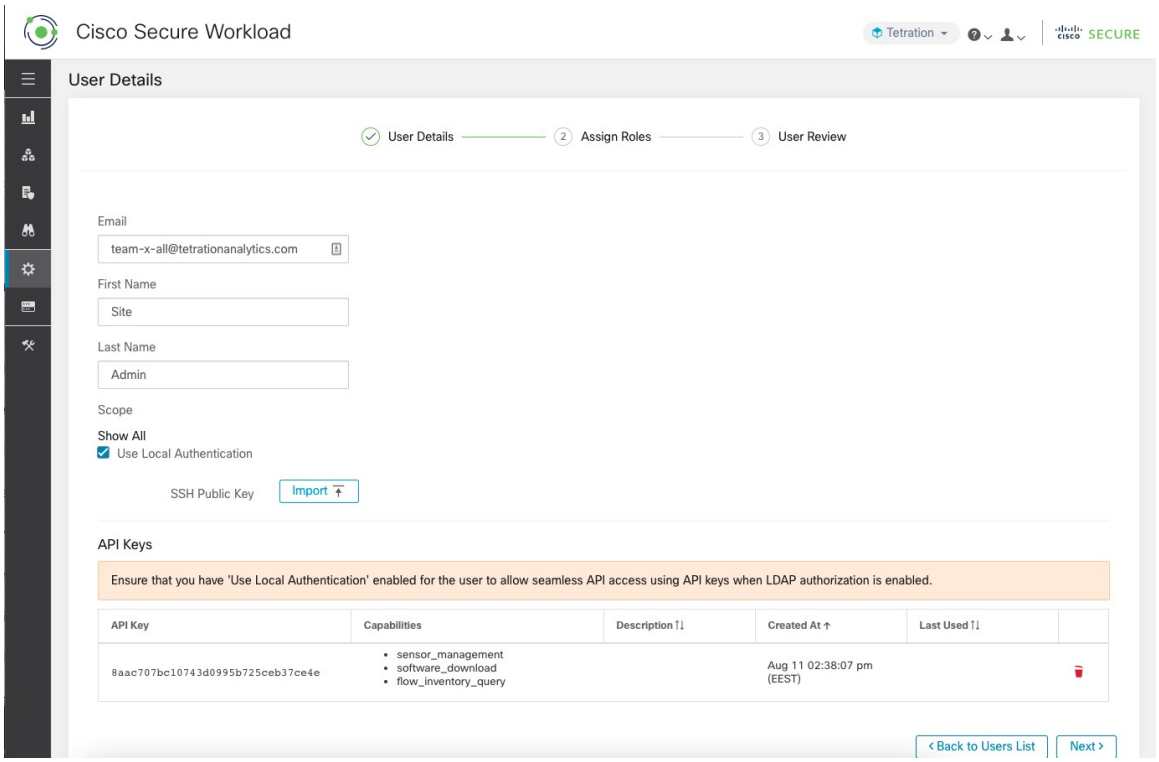


LDAP 承認が有効な場合、ユーザーセッションが終了すると LDAP MemberOf グループから派生した Secure Workload ロールが再評価されるため、API キーを介した OpenAPI へのアクセスはシームレスに機能しなくなります。したがって、中断のない OpenAPI アクセスを保証するために、API キーを持つすべてのユーザーが 「「Use Local Authentication」オプション」を有効にすることを推奨します。

図 *570* : *LDAP* 認証 *API* キーの警告



図 *571* : ユーザーページでの *LDAP* 認証 *API* キーの警告



LDAP 認証の問題のデバッグ

[外部認証（External Authentication）]、[LDAPグループからロールへのマッピング（LDAP Group to Role Mappings）] セクションで定義されたマッピングに基づいてロールがユーザーに割り当てられない場合は、ロールマッピングの設定と形式をもう一度確認してください。

- グループ文字列は文字列形式である必要があります。例：CN=group.jacpang,OU=Organizational,OU=Cisco Groups,DC=stage,DC=cisco,DC=com
- グループ名は、スペースや余分な文字が含まれず、AD に存在するものと正確に一致する必要があります。
- グループのロールマッピングは、ロールセレクタから選択する必要があります。

ユーザーロールマッピングのデバッグ手順

- 2 人のユーザーが必要です。1 人はサイト管理者（Site Admin）で、このユーザーの電子メールは AD ユーザーと同じであってはなりません。
- 以下の手順では、このユーザーを「SA ユーザー」と呼びます。
  - SA ユーザーには、前述のように、会社ページの外部認証構成でロールマッピング構成が事前に設定されています。「SA ユーザー」が [site-admin]@[ドメイン] でログインするとします。

- 「AD ユーザー」は [ad-user]@[Domain] であると仮定します。LDAP のセットアップが完了し、AD ユーザーはログインできますが、ロールを割り当てられていないと仮定します。

- AD ユーザーとして、シークレットブラウザセッションを使用してログインします。これにより、ブラウザの状態が SA ユーザーセッションから分離されます。
- SA ユーザーとしてログインし、[ユーザー（Users）] ページに移動します。
- ロールマッピングを構成する必要がある AD ユーザーの [編集（Edit）] アイコンをクリックします。
- [ユーザープロファイル（User Profile）] ページの [外部ユーザープロファイル（External User Profile）] ボタンをクリックします。
- 「memberof」セクションを含む外部認証プロファイルテーブルが表示されます。
- これは、会社ページ、[外部認証設定（External Authentication Config）]、[LDAPグループからロールへのマッピング（LDAP Group to Role Mappings）] セクションでのロールマッピングに使用できる「memberof」値の 1 つです。
- 「memberof」の行ごとの文字列全体を指定して一致させる必要があります。このロールマッピングを作成すると、同じ属性「memberof」を持つすべてのユーザーに、マップされたロールが割り当てられます。
- 新しくマップされたロールを AD ユーザーに付与するには、ユーザーはログアウトしてから再度ログインして、このマッピングプロファイルを再評価できるようにする必要があります。
- ユーザーがログインし、グループからロールへのマッピングの結果としてロールが正常に割り当てられると、一致するルールがそのユーザーの [設定（Preferences）] ページに表示されます。

シングルサインオン（SSO）の設定

このオプションを選択すると、SSO を使用してユーザーを認証できます。つまり、これを有効にすると、すべてのユーザーが認証のために ID プロバイダーのサインインページにリダイレクトされます。「Use Local Authentication」オプションが有効になっているユーザーは、サインインページで電子メールとパスワードのサインインフォームを使用して認証できます。

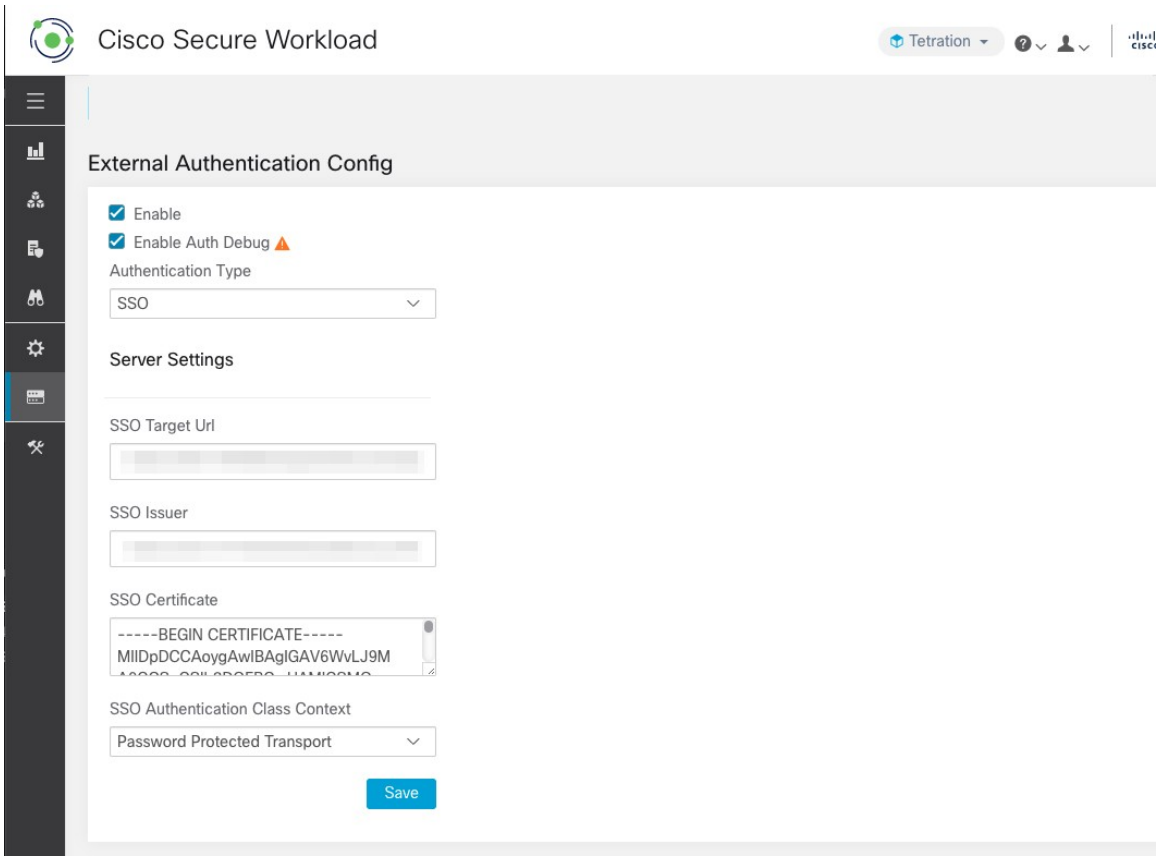
特に「Use Local Authentication」オプションが有効なユーザーがいない場合は、SSO が正しく設定されていることを確認することが重要です。推奨されるアプローチは、「Use Local Authentication」オプションをオンにして、**サイト管理者**のログイン情報を持つローカル認証されたユーザーを少なくとも 1 人指定する方法です。このユーザーは、SSO が正しく設定されていることを確認できます。接続が正常にセットアップされたら、ユーザー編集フローで [ローカル認証を使用（Use Local Authentication）] オプションをオフにして、このユーザーを外部認証に移行させることもできます。

SSOが有効になっている場合、新しいユーザーを作成するための推奨ワークフローは次のとおりです。

サイト管理者と**範囲所有者**は、新しいユーザーが SSO で初めてログインする前に、まず自分の電子メールで新しいユーザーを作成し、適切なロールと範囲を割り当てるようにお勧めします。新しいユーザーが適切なロールなしで SSO でログインした場合、デフォルトのロールはユーザーに割り当てられません。

次の表では、Secure Workload で SSO を設定するために設定する必要があるフィールドについて説明します。この場合の Secure Workload は、サービスプロバイダー（SP）です。

図 *572 :* シングル サインオンの設定



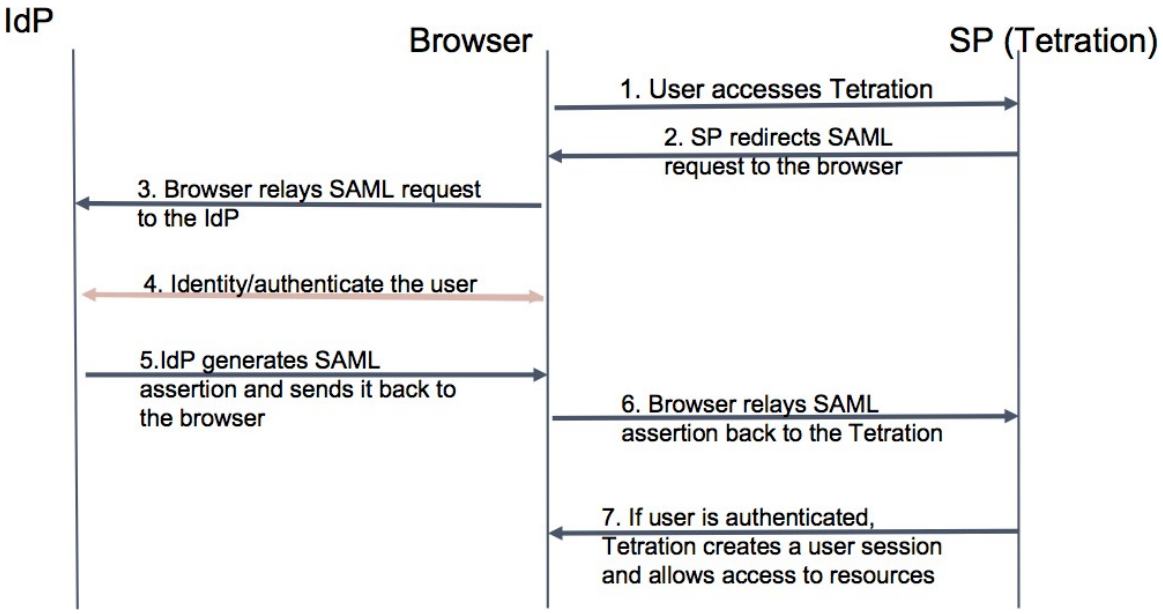
フィールド	説明
[SSOターゲットURL（SSO Target Url）]	サインインのためにユーザーがリダイレクトされる SSO IdP ターゲット URL。
[SSO発行元（SSO Is-suer）]	SP の SSO エンティティ ID、SP を一意に識別するための URL。これは通常、SP のメタデータです。このケースの場合： `https://<tetration-cluster-fqdn>/h4_users/saml/metadata`
[SSO証明書（SSO Certifi-cate）]	アイデンティティ プロバイダー（IdP）によって提供される SSO 証明書。

フィールド	説明
[SSO AuthNコンテキスト（SSO AuthN Context）]	SAML 要求で指定された SSO AuthN コンテキストの選択肢。デフォルトのオプションは [パスワードで保護されたトランスポート（Password Protected Transport）] です。他の選択肢は、Windows および PIV ベースの認証用の [統合Windows認証（Integrated Windows Authentication）] および [X.509証明書（X.509 Certificate）] です。

SSO 設定を有効にすると、[ローカル認証を使用（Use Local Authentication）] オプションが有効になっているユーザーを除いて、すべてのユーザーがセッションからログアウトされます。

[保存（Save）] ボタンをクリックすると、SSO 設定を保存できます。

図 *573 :* 認証ワークフロー



ID プロバイダー（IdP）に提供する情報

IdP は、認証用の SSO を設定するために Secure Workload（SP）からの情報を必要とします。次の表で、設定する必要があるフィールドについて説明します。

フィールド	説明
SSO URL（SSO Url）	SAML アサーション（IdP からの応答）を使用する認証エンドポイント（URL）。このケースの場合、次のようになります。 `https://<tetration-cluster-fqdn>/h4_users/saml/auth`
エンティティ ID（Entity Id）	これは SP のメタデータです。このケースの場合、次のようになります。 `https://<tetration-cluster-fqdn>/ h4_users/saml/metadata`

フィールド	説明
名前 ID の形式（Name ID Format）	名前 ID は電子メールアドレスです。 `'urn:oasis:names:tc:SAML:1. 1:nameid-format:emailAddress'`
属性（Attributes）	ユーザー属性はIdPから取得されます。シスコでは認証の一部としてこれらの属性を取得します。 • email • firstName • lastName 属性名が上記のとおりであることを確認してください。

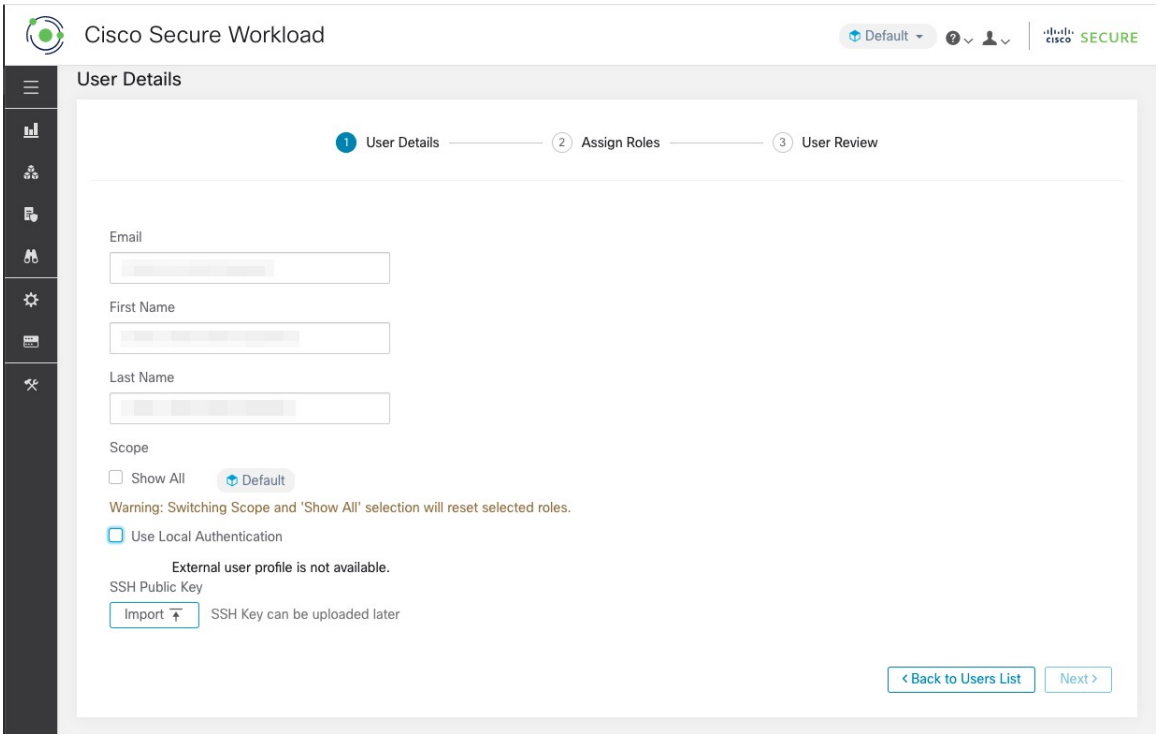
SSO の問題のデバッグ

- （サービスプロバイダーから）認証が機能することを確認できるのは設定後だけなので、SSO 構成の設定にはダウンタイムを設定します。
- 生成された IdP メタデータを確認して検証します。
- IdP と SP の間で交換されるすべての構成パラメータを確認します。
  - IdP での構成：SSO URL、対象者、名前 ID、属性など。
  - [Secure Workload Company] ページの設定：SSO ターゲット URL、SSO 発行者、および SSO 証明書。
- IdP から返されたサンプル SAML アサーションをサーバー アプリケーション ログから取得します。SAML バリデータに対して検証を行い、有効な SAML 応答であることを確認します。
- SP SSO セットアップでエラーが発生すると、IdP からエラーが生成される場合があります。ブラウザの Inspect 要素を使用すると、実行されているネットワークリクエストを確認できます。
- ユーザーのログインに問題がある場合は、Secure Workloadアプリケーションへのアクセス権をそのユーザーが持っているか IdP 管理者に確認してもらいます。

「Use Local Authentication」オプション

構成がセットアップされると、サイト管理者はユーザーが外部認証を使用しないようにできます。ユーザー編集セクションの「ローカル認証を使用」フラグを有効にすると、ユーザーごとに設定できます。ユーザーに対してこのフィールドを選択すると、そのユーザーはすべてのセッションからログアウトされます。

図 *574 : Use Local Authentication*



警告 **少なくとも 1 人のユーザーがローカル認証アクセスを持っていることを確認してください。**

ユーザーの「Use Local Authentication」オプションが削除されており（つまりチェックされていない）、たまたまこのユーザーがオプションを使用した最後のユーザーだった場合、Secure Workload にサインインするためのローカル認証アクセスを持つユーザーがいなくなります。つまり、設定や接続の問題など、外部認証システムに何らかの障害が発生した場合、サインインできるユーザーがいないということになります。ローカルで認証された最後のユーザーを削除しようとすると、警告が表示されます。

外部認証を介してログインするユーザーのセッションは短くなり、セッションの有効期限が切れるとログインするように求められます。外部認証を介してログインするユーザーは、サイトでパスワードをリセットできません（勤務先の Web サイトで行う必要があります）。ただし、ユーザーに「ローカル認証を使用」フラグが設定されている場合は、パスワードのリセットが可能です。

SSL 証明書およびキー

Secure Workload UI への完全に検証可能な HTTPS アクセスを有効にするには、UI のドメイン名に固有の SSL 証明書と、SSL 証明書の公開キーと一致する RSA 秘密キーをクラスタにアップロードします。

SSL 証明書は、Secure Workload UI 仮想 IP（VIP）アドレスを参照するために使用される完全修飾ドメイン名（FQDN）の形式に応じて、2 つの方法で取得できます。Secure Workload FQDN が Tetration.cisco.com などのエンタープライズドメイン名に基づいている場合、ベースドメインを所有するエンタープライズ認証局（CA）が SSL 証明書を発行します。それ以外の場合は、信頼できる SSL 証明書ベンダーを使用して、FQDN の SSL 証明書を発行できます。



（注） Secure Workload UI はサーバー名表示（SNI）をサポートしていますが、証明書で指定されたサブジェクトの代替名（SAN）は一致しないことに注意してください。たとえば、証明書の共通名（CN）が tetration.cisco.com であり、証明書にtetration1.cisco.com の SAN が含まれている場合、ホスト名はその証明書では提供されないため、HTTPS リクエストは SNI 互換ブラウザを使用して tetration1.cisco.com のクラスタに送信されます。CN で指定されたホスト名以外のホスト名でクラスタに対して行われた HTTPS リクエストは、クラスタにインストールされているデフォルトの自己署名証明書を使用して処理されます。これらのリクエストの結果、ブラウザに警告が表示されます。

サイト管理者と**カスタマーサポートのユーザー**は、SSL 証明書を使用できます。左側のナビゲーションバーで、**[プラットフォーム（Platform）]**>**[SSL証明書（SSL Certificate）]**をクリックします。

証明書とキーをインポートするには、[新しい証明書とキーのインポート（Import New Certifcate and Key）] ボタンをクリックします。



（注） SSL証明書と秘密キーの最初のインポートは、信頼ネットワーク接続を介してクラスタに対して実行し、トランスポート層にアクセスできる悪意のある第三者が秘密キーを傍受できないようにする必要があります。

SSL 証明書とキーについて、次の情報を入力します。

[名前（NAME）] : 証明書キーペアの任意の名前にできます。この名前は、インストールされている SSL 証明書を確認するときに役立ちます。

[X509証明書（X509 Certificate）] フィールドには、プライバシー強化メール（PEM）形式の SSL 証明書文字列を入力できます。SSL 証明書に中間 CA バンドルが必要な場合は、証明書の後に CA バンドルを連結して、Secure Workload FQDN の SSL 証明書が証明書ファイルの先頭になるようにします。

次の形式にする必要があります。

```
-----BEGIN CERTIFICATE-----

< Certificate for Secure Workload FQDN >

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

< Intermediary CA 1 content >

-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----

< Intermediary CA 2 content >

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

< Root CA content >

-----END CERTIFICATE-----
```

[RSA秘密キー（RSA Private Key）] フィールドには、前述の証明書で署名された公開キーの RSA 秘密キーを入力する必要があります。次の形式にする必要があります。

```
-----BEGIN RSA PRIVATE KEY-----

< private key data >

-----END RSA PRIVATE KEY-----
```



（注） RSA 秘密キーは暗号化されていない必要があります。RSA 秘密キーが暗号化されている場合、「500 内部サーバーエラー」が発生します。

インポートボタンを押すと、検証手順が実行され、証明書に署名された公開キーと秘密キーが実際にRSAキーペアであることが確認されます。検証に成功すると、証明書バンドルのSHA1ダイジェスト（SHA1 署名と作成時刻）が表示されます。

ブラウザをリロードして、Secure Workload UI への SSL 接続で新しくインポートされた SSL 証明書が使用されていることを確認します。

クラスタの設定

このセクションには、カスタマーネットワークおよび管理連絡先に関する Secure Workload クラスタの実行コンフィギュレーションが表示されます。編集可能な値は鉛筆アイコンで示されます。

（注） a. エージェント接続の強力な SSL 暗号：このオプションを有効にすると、 TLS-1.0 および TLS-1.1 プロトコルと次の暗号が、SSL ネゴシエーション中に Secure Workload クラスタによって受け入れられません。DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA256:DHE-RSA-AES256-SHA:ECDHE-ECDSA-DES-CBC3-SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:DES-CBC3-SHA

次の接続では、TLS ハンドシェイク中に強力な暗号が使用されます。

1. Secure Workload へのすべての API および UI 接続。
2. Secure Workload へのすべての可視性と適用エージェントの接続。

古い SSL ライブラリでは、このオプションがサポートされていない場合があります。

サイト管理者と**カスタマーサポートのユーザー**は、この設定にアクセスできます。左側のナビゲーションバーで、**[プラットフォーム（Platform）]**>**[クラスタ構成（Cluster Configuration）]** をクリックします。

構成の編集後、新しい構成がクラスタ全体に適用されるまでには時間がかかり、その間、特定の構成が強調表示されます。

外部 IPv6 クラスタの接続

物理 Cisco Secure Workload クラスタは、外部 IPv4 および IPv6 ネットワークの両方に接続するように設定できます。IPv4 接続は必須ですが、IPv6 接続は任意です。IPv6 接続は、一度設定すると無効化できません。クラスタの外部ネットワークの IPv6 接続は、展開またはアップグレード中にのみ有効にできます。アップグレード中に外部 IPv6 クラスタ接続を有効にする方法の詳細については、Cisco Secure Workload アップグレードガイド [英語] を参照してください。展開中に外部 IPv6 クラスタ接続を有効にする方法の詳細については、Cisco Secure Workload ハードウェア導入ガイド [英語] を参照してください。

始める前に

エージェントをデュアルスタックモード（IPv4 と IPv6 の両方をサポート）で動作させるには

前提条件

- クラスタでは IPv6 が有効になっている必要があります。
- FQDN の DNS に A および AAAA レコード（IPv4 および IPv6 用）を作成し、ドメイン名が解決されるまで待ちます。

エージェントがデュアルスタックモードで動作するように「センサー VIP FQDN」を設定する

手順

ステップ 1 左側のナビゲーションバーから、**[プラットフォーム（Platform）] > [クラスタ構成（Cluster Configuration）]**を選択します。

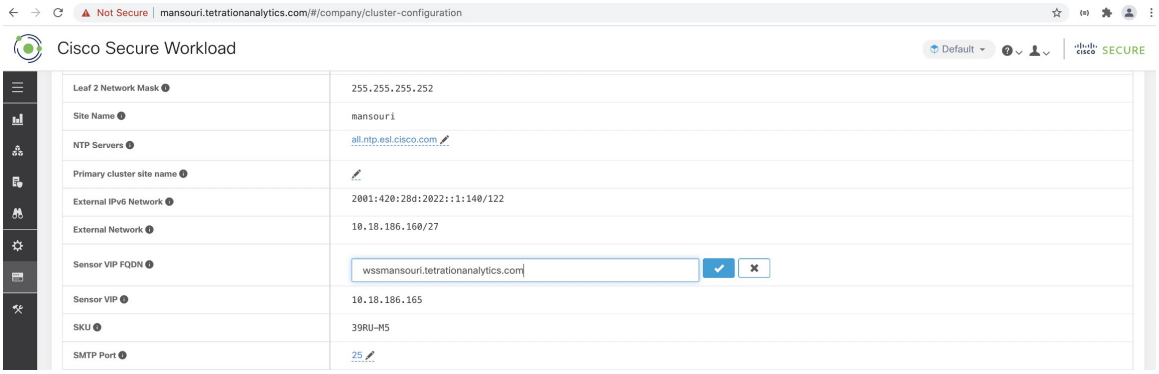
ステップ 2 [センサーIPv6 VIP（Sensor IPv6 VIP）]、[センサーVIP（Sensor VIP）]、および [センサーVIP FQDN（Sensor VIP FQDN）] フィールドを探します。[センサーIPv6 VIP（Sensor IPv6 VIP）] と [センサーVIP（Sensor VIP）] は設定されている必要があります。

ステップ 3 [センサーVIP FQDN（Sensor VIP FQDN）] が設定されていない場合は、前述の手順で作成した FQDN に設定します。設定する前に、FQDN の DNS の A レコードと AAAA レコードを解決する必要があります。

ステップ 4 [センサーVIP FQDN（Sensor VIP FQDN）] がすでに設定されている場合は、[センサーVIP FQDN（Sensor VIP FQDN）] フィールドで設定された FQDN の DNS に A レコードと AAAA レコードがあることを確認し、[センサーVIP FQDN（Sensor VIP FQDN）] フィールドをクリックして同じ値に保存し、更新されるようにします。

ステップ 5 フィールドの更新が完了すると（約20分後にステータスが自動的に更新されます）、エージェントは IPv4 と IPv6 の両方を介してクラスタに接続できるようになります。

ステップ 6 有効な [センサーVIP FQDN（Sensor VIP FQDN）] は 1 回だけ設定できます。



（注） AIX の IPv6 適用のサポートはありません。デュアルスタックモードの要件と制限の詳細については、Cisco Secure Workload アップグレードガイド [英語] を参照してください。

NTP 認証

NTP 認証は、Cisco Secure Workload オンプレミスバージョンでサポートされていて、展開中に設定 UI を使用するか、Cisco Secure Workload UI の [クラスタ設定（Cluster Configuration）] ページを使用して設定できます。

（注） Cisco Secure Workload は以下をサポートします。

- Network Time Protocol バージョン 4
- SHA1 認証

Cisco Secure Workload UI を使用して NTP 認証を設定するには、次の手順を実行します。

手順

ステップ 1 NTP サーバを設定します。次の設定は、CentOS 7 を実行しているシステムから参照用に提供されています。設定は、OS によって異なる場合があります。

a) 次のエントリが /etc/ntp.conf にあることを確認します。

```
# Key file containing the keys and key identifiers used when operating with symmetric
 key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
trustedkey 1
controlkey 1
requestkey 1
```

b) サーバー側のキーを /etc/ntp/keys に入力します。

```
# For more information about this file, see the man page ntp_auth(5).
# id type key
1 SHA1 <password>
```

c) NTP サーバーを再起動します。`# service ntpd restart`

d) サービスが開始されていることを確認します。

```
# ntpq -p
    remote       refid     st  t  when  poll  reach  delay  offset  jitter
==============================================================================
<ntp.server.com> <refid>     5  u  17     64     377  0.000  0.000   0.000
```

ステップ 2 Cisco Secure Workload UI の左側のナビゲーションウィンドウで、**[プラットフォーム（Platform）] > [クラスタ設定（Cluster Configuration）]** をクリックします。

ステップ 3 [認証済みNTPサーバー（Authenticated NTP Server）] フィールドに、NTP サーバーの名前または IP アドレスを入力します。

ステップ 4 [認証済みNTPサーバーのパスワード（Password For Authenticated NTP Server）] フィールドに、NTP サーバーのパスワードを入力します。

NTP サーバーが設定されて認証されると、認証済み NTP サーバーは、Cisco Secure Workload に入力されている認証されていない NTP サーバーよりも優先されます。

使用状況分析

シスコでは、Secure Workloadユーザーインターフェイスを改善するためにのみ使用するデータを収集しています。収集したデータは、サーバーに送信される前に一方向ハッシュによって匿名化されます。データ収集はデフォルトで有効になっており、このページで切り替えることができます。このプライバシー設定の構成可能性は、オンプレミスアプライアンスの場合はアプライアンス単位、Cisco Secure Workload SaaS の場合はテナント単位です。

サイト管理者と**カスタマーサポートのユーザー**は、使用状況分析を有効化または無効化できます。左側のナビゲーションバーで、**[管理（Manage）]** > **[使用状況分析（Usage Analytics）]**をクリックします。

[連携（Federation）]

（注） この機能を使用するには、フェデレーション内のすべてのアプライアンスがリリース 3.4.x 以降を実行している必要があります。

フェデレーションは、複数の Cisco Secure Workload アプライアンスを結合し、それらの管理の多くを**リーダー**として指定された単一のアプライアンスに統合する手段を提供します。

設定

フェデレーションを有効にするには、サイドバーの[会社（Company）]の下にある[フェデレーション（Federation）] メニュー項目を見つけます。

手順

ステップ 1 指定された**リーダー**で、**[会社（Company）]** > **[フェデレーション（Federation）]** に移動し、[新しいフェデレーションの作成（Create New Federation）] ボタンをクリックします。

ステップ 2 最初の**フォロワー**アプライアンスを追加するために、その名前と完全修飾ドメイン名（FQDN）を入力し、[追加（Add）] ボタンをクリックします。

ステップ 3 リンクをクリックして、参加証明書ファイルをダウンロードします。

ステップ 4 **フォロワー**で、**[会社（Company）]** > **[フェデレーション（Federation）]** に移動し、[既存のフェデレーションへの参加（Join Existing Federation）] をクリックして、上記で作成した参加証明書を選択します。

ステップ 5 フェデレーションの一部となる**フォロワー**ごとに、ステップ 2 ～ 4 を繰り返します。

図 *575* : フェデレーションの作成またはフェデレーションへの参加

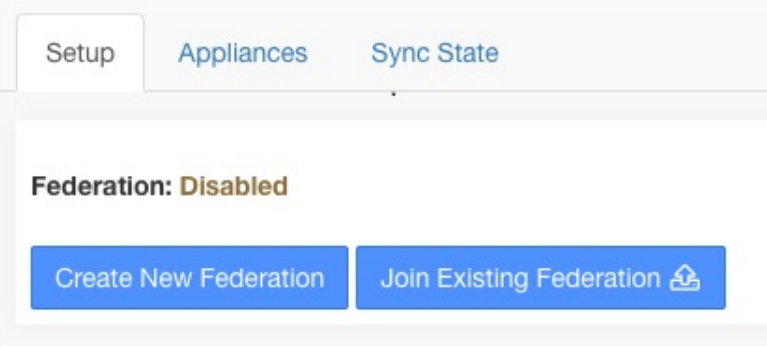
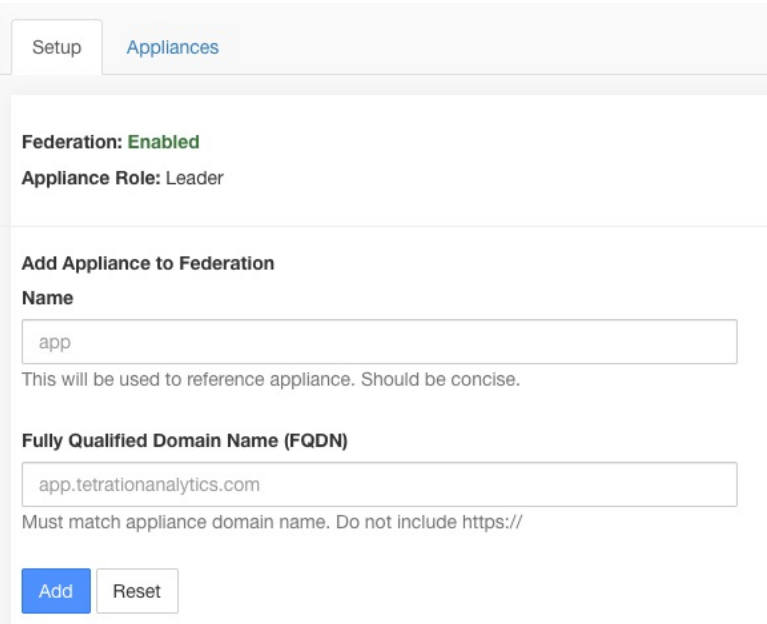
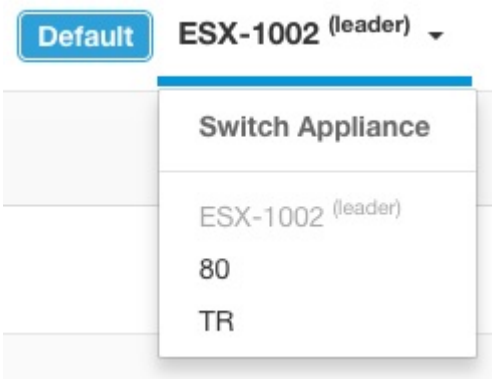


図 *576* : フェデレーションのフォロワーの追加フォーム



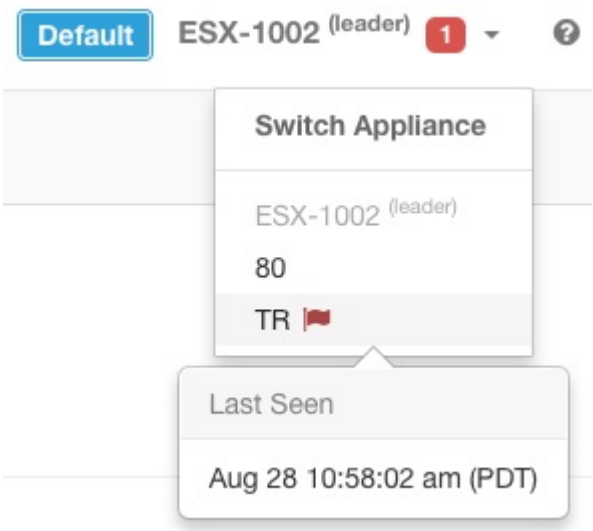
フェデレーションが有効になると、ヘッダーにはアプライアンスの名前と、アプライアンスを変更するためのセレクタが含まれます。

図 *577* : アプライアンスセレクタ



フェデレーション内の 1 つ以上のアプライアンスが 10 分間以上リーダーから認識されなかった場合、アプライアンスセレクタにアラートが表示され、問題のあるアプライアンスにフラグが付けられます。カーソルを合わせると、最後にリーダーと同期した時刻が表示されます。

図 *578* : アラートが表示されたアプライアンスセレクタ



認証設定

フェデレーションが有効になっている認証は、シングルサインオン（SSO）を使用して設定されます。SSOは、フェデレーションの各アプライアンス部分で設定する必要があります。SSOの設定は、各アプライアンスでのシングルサインオン（SSO）の設定で示されているように、企業ページの外部認証設定を介してリーダーと各フォロワーで設定されます。

管理タスク

管理タスクに応じて、**リーダー**で実行する必要があるタスクと**フォロワー**で実行する必要があるタスクがあります。次の表に、各タスクのアプライアンスタイプを示します。

タスク	アプライアンス
Users	Leader
スコープ	Leader
ロール（Roles）	Leader
テナント	Leader
API キー	Leader
[収集ルール（Collection Rules）]	Leader
ソフトウェアエージェントの設定	Leader
ソフトウェアエージェント	フォロワー
ソフトウェアエージェントのアップグレード（Software Agent Upgrade）	フォロワー
ソフトウェアエージェントのダウンロード（Software Agent Download）	フォロワー
[インベントリフィルタ（Inventory Filters）]	Leader
インベントリアップロード（Inventory Upload）	Leader
デフォルトのポリシー検出設定	Leader
ポリシーの順序	Leader

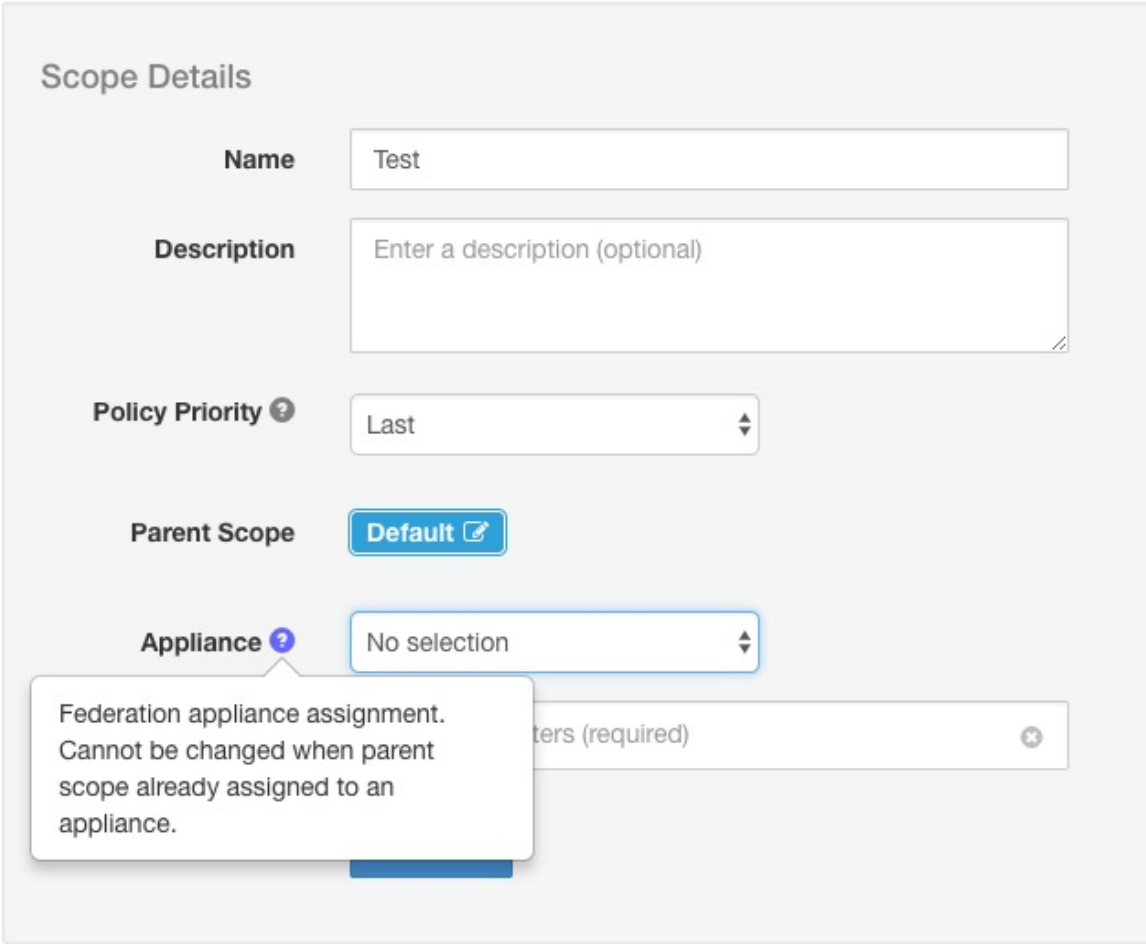
スコープ

範囲内のすべてのインベントリが単一のアプライアンスによって管理されている場合、その範囲をアプライアンスに割り当てることができます。これにより、その範囲に関連付けられたワークスペースでの自動ポリシー検出、ポリシー分析、および適用が有効になります。また、その範囲で作成されたポリシーが、アプライアンスに接続されているエージェントにのみ適用されるようになります。

（アプライアンスに割り当てられていない）グローバル範囲で作成されたアプリケーションは、自動ポリシー検出またはポリシー分析には使用できません。ただし、フェデレーション内のすべてのアプライアンスにポリシーを適用するために使用できます。

アプライアンスは、作成時に、または範囲を編集して、範囲に割り当てることができます。すべての子範囲は親のアプライアンスを継承し、別のアプライアンスに割り当てることはできません。

図 *579*：範囲へのアプライアンスの割り当て



（注） ルートレベルの範囲（テナント）は常にグローバルであり、アプライアンスに割り当てることはできません。

ワークスペース

すべてのワークスペース（「アプリケーション」）は、**リーダー**で管理する必要があります。ただし、フローベースのチャートは、対応する**フォロワー**アプライアンスでのみ表示できます。これには、[ポリシー分析（Policy Analysis）] タブと [適用（Enforcement）] タブに表示されるチャートが含まれます。**リーダー**から、[ローカルアプライアンスでチャートを表示（View Charts on Local Appliance）] をクリックして、対応する**フォロワー**に移動します。

図 *580* : リーダーでのポリシー分析

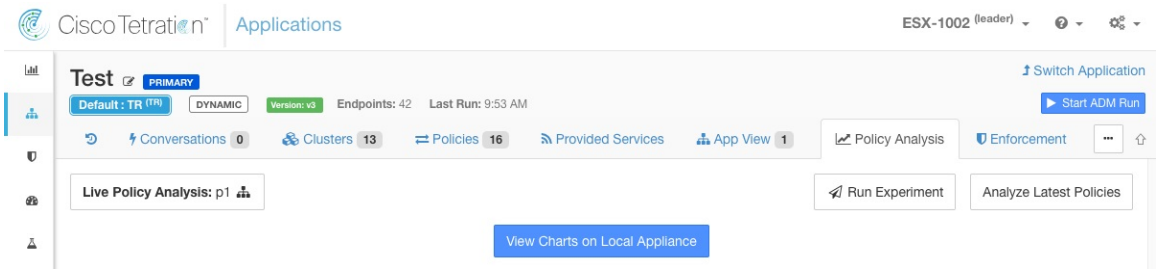
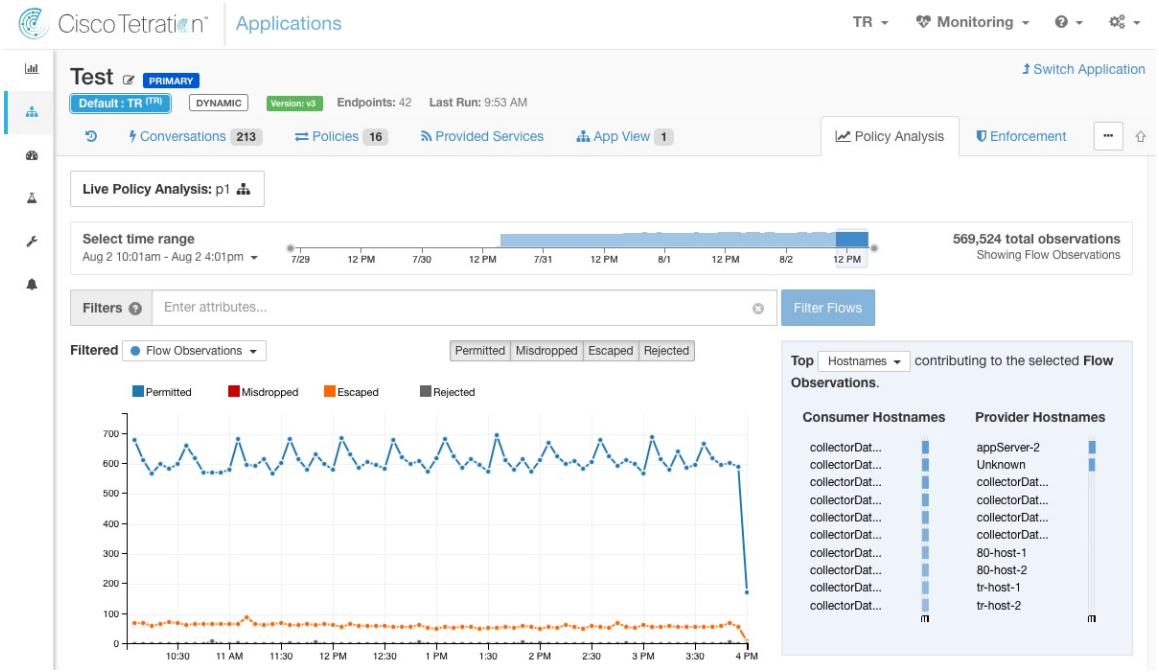
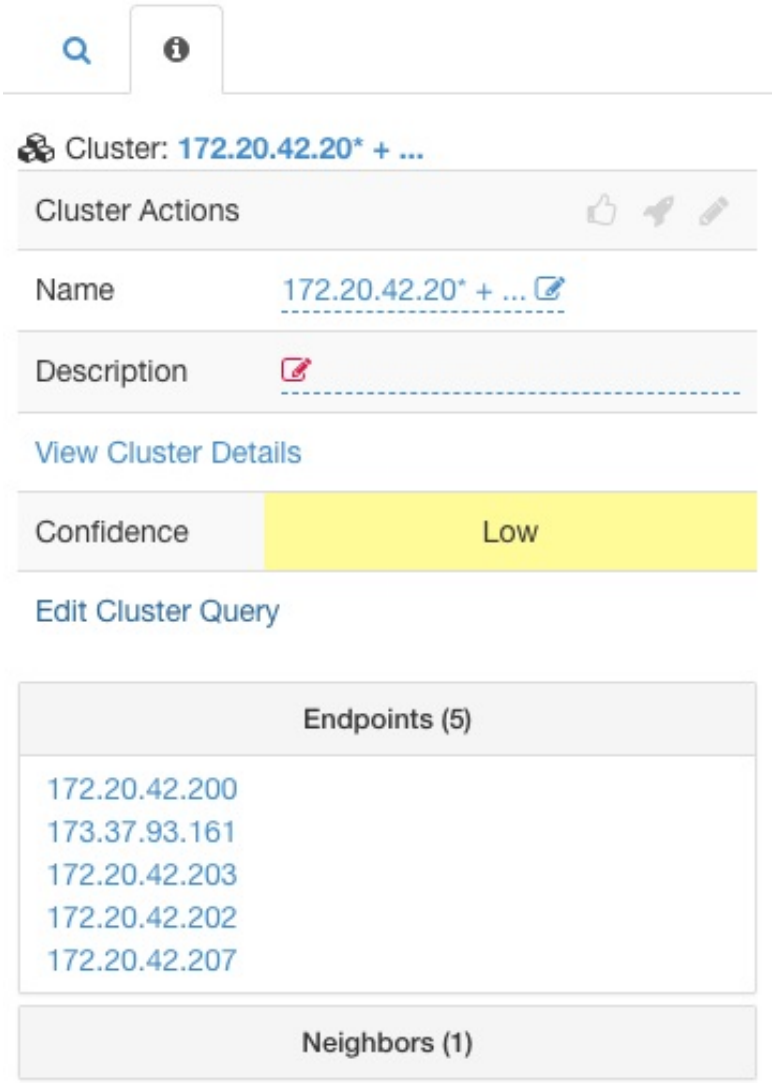


図 *581* : フォロワーでのポリシー分析



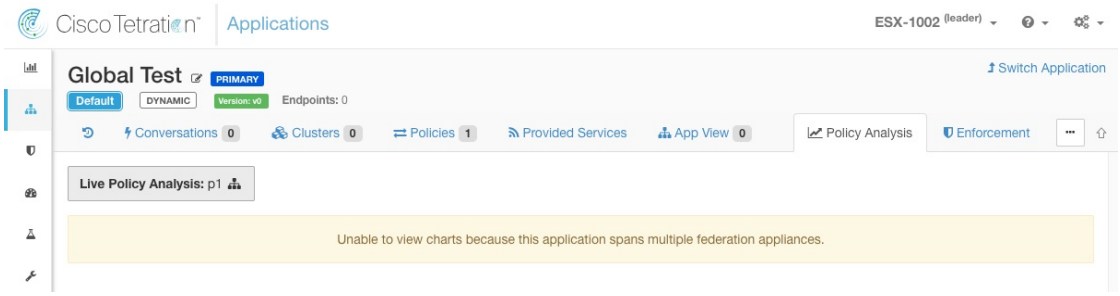
また、インベントリに対する検索（自動ポリシー検出ページを除く）は、常にローカルで実行されます。そのため、クラスタ、フィルタ、および範囲のエンドポイントを表示するには、**フォロワー**に移動する必要があります。クラスタ、フィルタ、および範囲の詳細の表示にも同じロジックが適用されます。

図 *582 :* クラスタサイドバー



前述のように、グローバル範囲で作成されたワークスペースは、自動ポリシー検出またはポリシー分析には使用できません。ポリシーは適用できますが、フローベースの適用チャートは使用できません。

図 *583 :* グローバル範囲で無効になっているポリシー分析



警告 アプライアンスに関連付けられた範囲または制限付きインベントリフィルタを使用するポリシーは、そのアプライアンスにのみ適用されます。

ソフトウェアエージェント

フェデレーション内のアプライアンスに接続されているすべてのソフトウェアエージェントが**リーダー**に表示されます。

手順

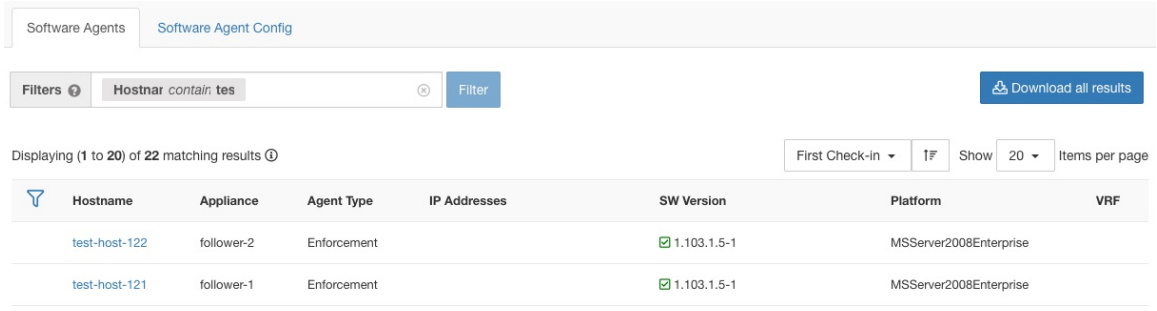
ステップ 1 右上隅にある [設定（Settings）] メニューをクリックします。

ステップ 2 [エージェント設定（Agent Config）] を選択します。
[エージェント設定（Agent Config）] ページが表示されます。

ステップ 3 [ソフトウェアエージェント（Software Agents）] タブをクリックします。
[ソフトウェアエージェント（Software Agents）] タブが開きます。

ステップ 4 移動するエージェントを見つけて、そのテーブル行にあるチェックボックスをクリックします。

ステップ 5 [アプライアンス（Appliance）] 列は、エージェントが接続されている場所を示しています。



フォロワーアプライアンス間でのソフトウェアエージェントの移動

ソフトウェアエージェントは、**フォロワー**アプライアンス間で移動できます。エージェントが接続されているアプライアンスから、次の手順を実行します。

手順

ステップ 1 右上隅にある [設定（Settings）] メニューをクリックします。

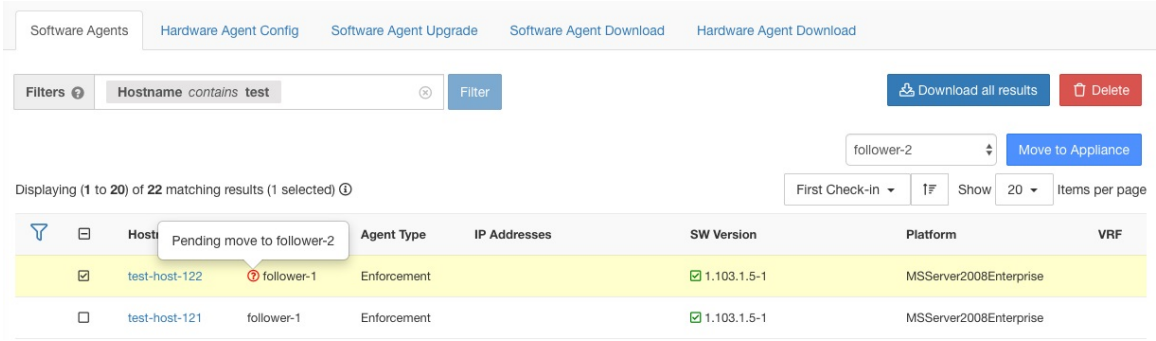
ステップ 2 [エージェント設定（Agent Config）] を選択します。ページが表示されます。

ステップ 3 [ソフトウェアエージェント（Software Agents）] タブをクリックします。[ソフトウェアエージェント（Software Agents）] タブが開きます。

ステップ 4 移動するエージェントを見つけて、そのテーブル行にあるチェックボックスをクリックします。

ステップ 5 [–アプライアンスの選択–（–Select Appliance–）] ドロップダウンから、これらのエージェントに必要なアプライアンスを選択します。

ステップ 6 [アプライアンスに移動（Move to Appliance）] ボタンをクリックします。



テーブルが更新され、移動が保留中であることが示されます。エージェントの次回チェックイン時に、アプライアンスを移動するためのメッセージが受信されます。移動が完了すると、エージェントは元のアプライアンスで表示されなくなります。新しいアプライアンスの [ソフトウェアエージェント（Software Agents）] ページにアクセスして、移動が成功したことを確認します。

その他のタスク

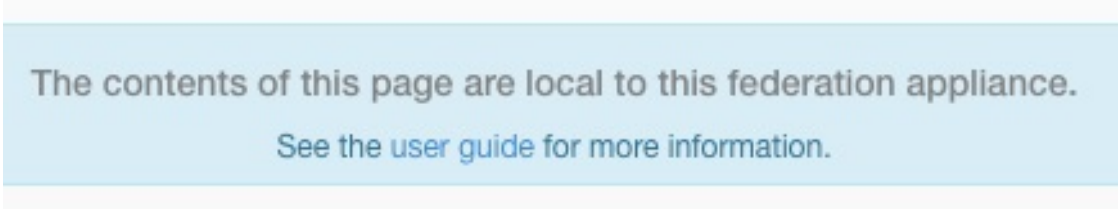
一般的に、フローおよびインベントリベースのクエリは、**フォロワー**アプライアンスで行う必要があります。次の表に、いくつかの一般的なタスクのアプライアンスタイプを示します。

タスク	アプライアンス
[可視性（Visibility）] > > **[フロー検索（Flow Search）]**	フォロワー

タスク	アプライアンス
[可視性（Visibility）] > [インベントリ検索（Inventory Search）]	フォロワー
[可視性（Visibility）] > [インベントリフィルタ（Inventory Filters）]	フォロワー
[可視性（Visibility）] > [外部オーケストレータ（External Orchestrators）]	フォロワー
[セグメンテーション（Segmentation）] > [自動ポリシー検出（Automatic Policy Discovery）]	Leader
[セグメンテーション（Segmentation）] > [ポリシー分析（Policy Analysis）]	Leader
[セグメンテーション（Segmentation）] > [適用履歴（Enforcement History）]	Leader
[セグメンテーション（Segmentation）] > [カンバセーション（Conversations）]	フォロワー
[セグメンテーション（Segmentation）] > [分析結果（Analysis Results）]	フォロワー
[セグメンテーション（Segmentation）] > [適用結果（Enforcement Results）]	フォロワー
[モニタリング（Monitoring）] > [エージェント（Agents）]	フォロワー
[モニタリング（Monitoring）] > [適用ステータス（Enforcement Status）]	フォロワー
[モニタリング（Monitoring）] > [ライセンス（Licenses）]	フォロワー
[ソフトウェアエージェント（Software Agents）] > [アプライアンスの変更（Change Appliances）]	フォロワー

上記に含まれていないその他のタスクは、アプライアンスに対してローカルであると考える必要があります。そのため、行われた変更または表示される結果は、現在のアプライアンスの状態のみを表し、フェデレーションは表しません。これらのページには、次のアラートが表示されます。

図 *584 :* ローカル アプライアンス アラート



既存の展開

このドキュメントでは、フェデレーションに参加するアプライアンスでデータを保持するための一連のガイドラインについて説明します。

保持できるデータ

ユーザーは、フェデレーションに追加する前に、ユーザー、ロール、収集ルール、フォレンジックプロファイル、ユーザーアップロードラベル、およびエージェント設定をフォロワーからリーダーにコピーする必要があります。リーダーにコピーされなかったフォロワーのデータは消去され、リーダーのデータに置き換えられます。

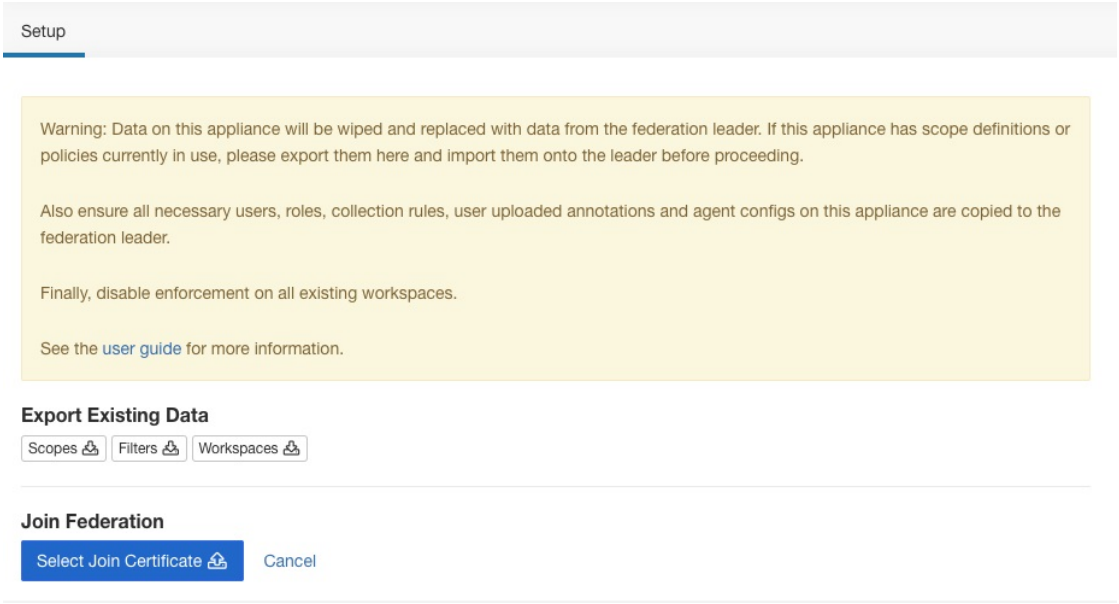
フォロワーの範囲、フィルタ、およびポリシーを保持するためにファイルにエクスポートし、**リーダー**にインポートできるようにするメカニズムを提供しています。これは、次のように行います。

手順

ステップ 1 フォロワーアプライアンスで、**[会社（Company）]** > **[フェデレーション（Federation）]** に移動し、[新しいフェデレーションへの参加（Join New Federation）] ボタンをクリックします。

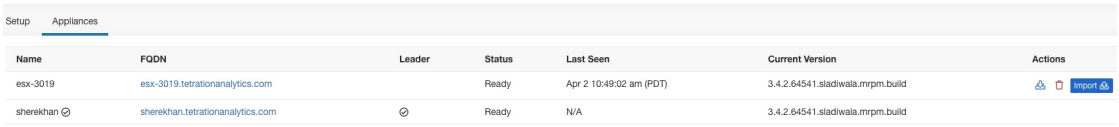
範囲、フィルタ、およびワークスペースをアプライアンスにローカルに [ダウンロード（Download）] します。

図 *585 :* フォロワーの既存の展開のエクスポートワークフロー



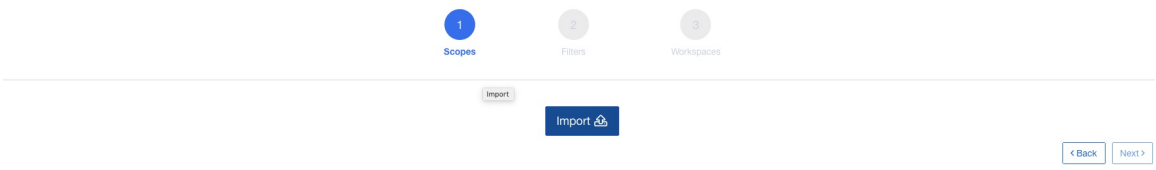
ステップ 2 **リーダー**で、**[会社（Company）]** > **[フェデレーション（Federation）]** に移動します。フォロワーの名前と完全修飾ドメイン名（FQDN）を入力し、[追加（Add）] ボタンをクリックしてフォロワーを追加します。次に、アプライアンスビューに切り替え、アプライアンスのFQDNの右側にある**インポートボタンをクリックします**。

図 *586 :* フォロワーの既存の展開のインポートアイコン



これにより、フォロワーからダウンロードした範囲、フィルタ、およびワークスペースを**アップロード**するようにユーザーに求めるインポートウィザードが表示されます。すべての段階で、ウィザードはユーザーが競合を解決するのを待ってから次の段階に進みます。

図 *587 :* フォロワーの既存の展開のインポートウィザード



リーダーとフォロワーのエントリ間の競合は、2 つのアプライアンスでこれらのエントリの名前を比較することで検出されます。たとえば、リーダーとフォロワーの両方に存在する範囲 **Default:host** があるとします。リーダーではこの範囲のクエリが **Hostname eq foo** に設定され、フォロワーでは **Hostname eq bar** に設定されています。インポートウィザードは、この範囲に

競合が存在することをユーザーに警告し、リーダーからのクエリ（つまり、**Hostname eq foo**）を選択します。

ステップ 3 最後に、ユーザーはフォロワーをフェデレーションに追加する前に、フォロワーの既存のすべてのワークスペースで**適用を無効にする**必要があります。

ステップ 4 フェデレーションに参加するフォロワーごとに、ステップ 1 ～ 3 を繰り返す必要があります。

保持できないデータ

1. 仮想アプライアンス（コネクタで使用されるものを含む）は、フェデレーションに参加させた後に、フォロワーで再プロビジョニングする必要があります。
2. フォロワーがフェデレーションに参加するまでのフォロワーでのフローデータは、リーダーと共通の範囲ではアクセスできません。

操作の切断モード

（注） フォロワーに適用できます。

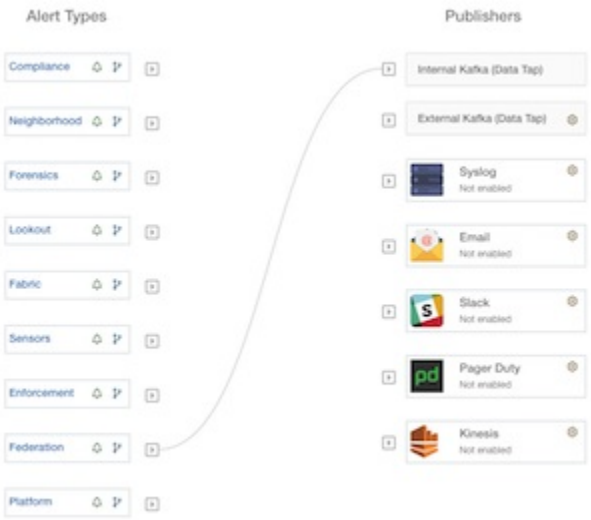
ネットワークパーティションなどの特定の状況では、1 つ以上の**フォロワー**でフェデレーションを無効にして、スタンドアロンモードで動作できるようにすることが理にかなっています。これを行うには、[会社（Company）] > [フェデレーション（Federation）] に移動し、[無効（Disable）] ボタンをクリックします。フェデレーションから切断されたフォロワーは、スタンドアロンクラスタとして動作を続けます。

フォロワーの新しい範囲、インベントリフィルタ、およびワークスペースは、ファイルにエクスポートすることで保持できます。このファイルが、フォロワーをフェデレーションに再度追加する前にリーダーでインポートされます。これにより、既存のワークスペースのポリシーへの変更が保持されます。ただし、フォロワーがフェデレーションに再参加すると、リーダーにすでに存在する範囲とインベントリフィルタへの変更は失われます。

アラート

アラートを有効にするには、**[アラート（Alert）] > [設定（Configuration）]** をクリックし、フェデレーションのアラート設定を更新します。

図 *588 :* フェデレーションアラート



アラートは、次のイベントに対して生成されます。

- フェデレーション内の 1 つ以上のアプライアンスが 10 分以上通信されていない場合、[中（MEDIUM）] の重大度でリーダーに対してアラートが生成されます。
- フォロワーが 10 分以上リーダーと通信できない場合、[中（MEDIUM）] の重大度でフォロワーに対してアラートが生成されます。

アラート詳細

一般的なアラート構造とフィールドに関する情報については、一般的なアラート構造（803 ページ）を参照してください。alert_details フィールドは構造化されており、フェデレーションアラートの次のサブフィールドが含まれています。

（注） アプライアンスは、アラートをトリガーしたアプライアンスです。

フィールド	アラートタイプ	フォーマット	説明
id	*all*	string	アプライアンスID
name	*all*	string	アプライアンス名
fqdn	*all*	string	アプライアンスの FQDN
is_leader	*all*	boolean	アプライアンスがリーダーの場合は True

フィールド	アラートタイプ	フォーマット	説明
status	*all*	string	アプライアンスのステータス
current_sw_version	*all*	string	アプライアンスのソフトウェアバージョン
last_seen_at	*all*	integer	アプライアンスが前回検出されたときの UNIX タイムスタンプ
created_At	*all*	integer	アプライアンスが作成されたときの Unix タイムスタンプ
updated_at	*all*	integer	アプライアンスが更新されたときの Unix タイムスタンプ
created_At	*all*	integer	アプライアンスが作成されたときの Unix タイムスタンプ
deleted_at	*all*	integer	アプライアンスが削除されたときの Unix タイムスタンプ
disconnected	*all*	boolean	フォロワーがリーダーから切断されている場合は true に設定されます。リーダーの場合は常に false に設定されます

フォロワーダウンアラートの **alert_details** の例

```
{
  "id": "5f219ad8755f024b46c2524a",
  "name": "esx-3018",
  "fqdn": "esx-3018.tetrationanalytics.com",
  "is_leader": false,
  "status": "Ready",
  "current_sw_version": "3.4.0.39.devel",
  "last_seen_at": 1596140582,
  "created_at": 1596037848,
  "updated_at": 1596140582,
  "deleted_at": 0,
  "disconnected": true
}
```

リーダーダウンアラートの **alert_details** の例

```
{
  "id": "5f219acc755f024b46c25248",
  "name": "sherekhan",
  "fqdn": "sherekhan.tetrationanalytics.com",
  "is_leader": true,
  "status": "Ready",
  "current_sw_version": "3.4.0.39.devel",
  "last_seen_at": 1596140582,
  "created_at": 1596037848,
  "updated_at": 1596140582,
  "deleted_at": 0,
  "disconnected": false
}
```


API



（注） フェデレーションクラスタのログイン情報は常にリーダーで生成する必要があり、フォロワーのクエリに使用できます。

この項では、フェデレーション用に追加または更新された API を示します。

アプライアンス

アプライアンスエンドポイントを使用すると、ユーザーはフェデレーション内のアプライアンスの状態を取得できます。

アプライアンスオブジェクト

アプライアンスオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	アプライアンスの固有識別子。
name	string	ユーザーが指定したアプライアンスの名前。
fqdn	string	ユーザーが指定したアプライアンスの FQDN。
is_leader	boolean	アプライアンスがリーダーであるかどうかを示します。
status	string	アプライアンスのステータス。
current_sw_version	string	アプライアンスの Cisco Secure Workload ソフトウェアのバージョン。

属性	タイプ	説明
last_seen_at	integer	フォロワーがリーダーにより最後に確認されたときの Unix タイムスタンプ。リーダーの場合は常に null です。
deleted_at	整数	アプライアンスが削除されたときの Unix タイムスタンプ。
disconnected	boolean	フォロワーがリーダーとの接続を失ったかどうかを示します。リーダーの場合は false に設定されます。

アプライアンスのリスト

このエンドポイントは、フェデレーション内のアプライアンスの配列を返します。

```
GET /openapi/v1/appliances
```

パラメータ：（なし）

応答オブジェクト：アプライアンスオブジェクトの配列を返します。

サンプル python コード

```
restclient.get('/appliances')
```

スコープ

範囲オブジェクト（1121 ページ）に、範囲に関連付けられたアプライアンスの ID が含まれるようになりました。グローバル範囲の場合は null に設定されます。

次の API が、範囲の作成または更新時にアプライアンス ID を受け入れるようになりました。

範囲の作成

範囲の作成時に指定されたアプライアンス ID により、範囲が特定のアプライアンスに関連付けられます。

```
POST /openapi/v1/app_scopes
```

パラメータ：

名前	タイプ	説明
short_name	string	ユーザーが指定した範囲の名前。
説明	string	ユーザーが指定した範囲の説明。

名前	タイプ	説明
short_query{1}JSON{1}	JSON	範囲に関連付けられているフィルタ (または一致基準)。
parent_app_scope_id	string	親範囲の ID。
policy_priority	integer	デフォルトは「last」です。ワークスペースの優先順位を並べ替えるために使用されます。自動検出されたポリシーの確認（633ページ）の「ポリシーの順序付け」を参照してください。
appliance_id	string	アプライアンスの固有識別子。

サンプル python コード

```
req_payload = {
    "short_name": "App Scope Name",
    "short_query": {
        "type":"eq",
        "field":"ip",
        "value": <....>
    },
    "parent_app_scope_id": <parent_app_scope_id>,
    "appliance_id": <appliance_id>,
}
resp = restclient.post('/app_scopes', json_body=json.dumps(req_payload))
```

範囲の更新

この API を使用すると、アプライアンス ID を使用して既存の範囲をアプライアンスに関連付けることができます。

```
PUT /openapi/v1/app_scopes/{app_scope_id}
```

パラメータ：

名前	タイプ	説明
short_name	string	ユーザーが指定した範囲の名前。
説明	string	ユーザーが指定した範囲の説明。
short_query{1}JSON{1}	JSON	範囲に関連付けられているフィルタ (または一致基準)。
appliance_id	string	アプライアンスの固有識別子。

指定された ID に関連付けられている変更された範囲オブジェクトを返します。

サンプル python コード

```
req_payload = {
    "short_name": "App Scope Name",
    "short_query": {
        "type":"eq",
        "field":"ip",
        "value": <....>
    },
    "appliance_id": <appliance_id>,
}
resp = restclient.put('/app_scopes/%s' % <app_scope_id>,
                      json_body=json.dumps(req_payload))
```

アイドルセッション

このセクションでは、ローカルデータベースを使用して認証を行う場合に、ログイン試行の失敗によってユーザーアカウントがどのようにロックされるかについて説明します。

手順

ステップ 1 電子メールアドレスとパスワードを使用したログイン試行が 5 回失敗すると、アカウントがロックされます。

（注） プロービングに対するセキュリティ対策として、ロックされたアカウントにサインインを試みた場合、ロックを示す具体的なメッセージはログインインターフェイスに表示されません。

ステップ 2 ロックアウト間隔は 30 分に設定されます。アカウントのロックが解除されたら、正しいパスワードを使用してログインするか、[パスワードを忘れた場合（Forgot password?）] をクリックしてパスワードの回復を開始します。

（注） 正常にサインインしたユーザーは、1 時間何も操作しないとログアウトされます。このタイムアウトは、**[管理（Manage）]**>**[セッション構成（Session Configuration）]** で設定します。

設定

[設定（Preferences）] ページにはアカウントの詳細が表示され、表示設定の更新、ランディングページの変更、パスワードの変更、および 2 要素認証の設定を行うことができます。

ランディングページ設定の変更

サインイン時に表示されるページを変更するには：

手順

ステップ 1 ウィンドウの右上隅にあるユーザーアイコンをクリックし、[ユーザー設定（User preferences）] を選択します。

ステップ 2 ドロップダウンメニューからランディングページを選択します。設定は、ログイン時のデフォルト/ホームページとして保存されます。変更を確認するには、ページの左上隅にある Secure Workload ロゴをクリックします。

パスワードの変更

手順

ステップ 1 右上隅にあるユーザーアイコンをクリックします。

ステップ 2 [ユーザー設定（User Preferences）] をクリックします。

ステップ 3 [パスワードの変更（Change Password）] ペインで、 [古いパスワード（Old Password）] フィールドに現在のパスワードを入力します。

ステップ 4 [パスワード（Password）] フィールドに新しいパスワードを入力します。

ステップ 5 [パスワードの確認（Confirm Password）] フィールドに新しいパスワードを再度入力します。

ステップ 6 変更を送信するには、[パスワード変更（Change Password）] をクリックします。

（注） **パスワードは 8 ～ 128 文字で、以下の文字記号を少なくとも 1 つ含める必要があります。**

- アルファベット小文字（a b c d. . .）
- アルファベット大文字（A B C D. . .）
- 数字（0 1 2 3 4 5 6 7 8 9）
- 特殊文字（ ! " # $ % & ’ ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ‘ { | } ~) スペースを含む

パスワードの回復

このセクションでは、パスワードを回復する方法について説明します。

始める前に

パスワードをリセットするには、まずアカウントを取得する必要があります。**サイト管理者**および**カスタマーサポートユーザー**は、新しいアカウントを追加できます。

手順

ステップ 1 ブラウザで Cisco Secure Workload URL にアクセスし、[パスワードを忘れた場合（Forgot Password）] リンクをクリックします。[パスワードをお忘れですか?（Forgot your password?）] ダイアログが表示されます。

ステップ 2 [電子メール（Email）] フィールドに電子メールアドレスを入力します。

ステップ 3 [Reset Password] をクリックします。

パスワードのリセット手順がメールに送信されます。

（注） 電子メールベースのパスワード回復にはワンタイムパスワードを含めることができないため、二要素認証のパスワード回復手順では、Secure Workload カスタマーサポートに連絡する必要があります。

二要素認証の有効化

このセクションでは、二要素認証を有効にする方法について説明します。

手順

ステップ 1 右上隅にあるユーザーアイコンをクリックします。

ステップ 2 [ユーザー設定（User Preferences）] をクリックします。

ステップ 3 [二要素認証（Two-Factor Authentication）] ペインで、[有効にする（Enable）] ボタンをクリックします。新しい [二要素認証（Two-Factor Authentication）] ペインが表示されます。

ステップ 4 パスワードを入力します。

ステップ 5 Google Authenticator（Android または iOS の場合）または Authenticator（Windows Phone の場合）などの時間ベースのワンタイムパスワード（TOTP）アプリケーションを使用して、[現在のパスワード（Current Password）] フィールドの下に表示されている QR コードをスキャンします。

ステップ 6 選択した TOTP アプリケーションによって表示される検証コードを入力します。

ステップ 7 [有効（Enable）] をクリックします。

図 *589 : [*二要素認証（*Two-Factor Authentication*）*]* ペイン



次にシステムにログインする際には、[二要素認証を使用する（Use two-factor authentication）] チェックボックスをオンにして、TOTP アプリケーションに表示される確認コードを入力してサインインする必要があります。

（注） 電子メールベースのパスワード回復にはワンタイムパスワードを含めることができないため、二要素認証のパスワード回復手順では、Secure Workload カスタマーサポートに連絡する必要があります。

二要素認証の無効化

このセクションでは、二要素認証を無効にする方法について説明します。

手順

ステップ 1 右上隅にあるユーザーアイコンをクリックします。

ステップ 2 [ユーザー設定（User Preferences）] をクリックします。

ステップ 3 二要素認証で、[無効にする（Disable）] ボタンをクリックします。[二要素認証（Two-Factor Authentication）] ペインが表示されます。

ステップ 4 パスワードを入力します。

ステップ 5 [無効にする（Disable）] ボタンを再度クリックします。

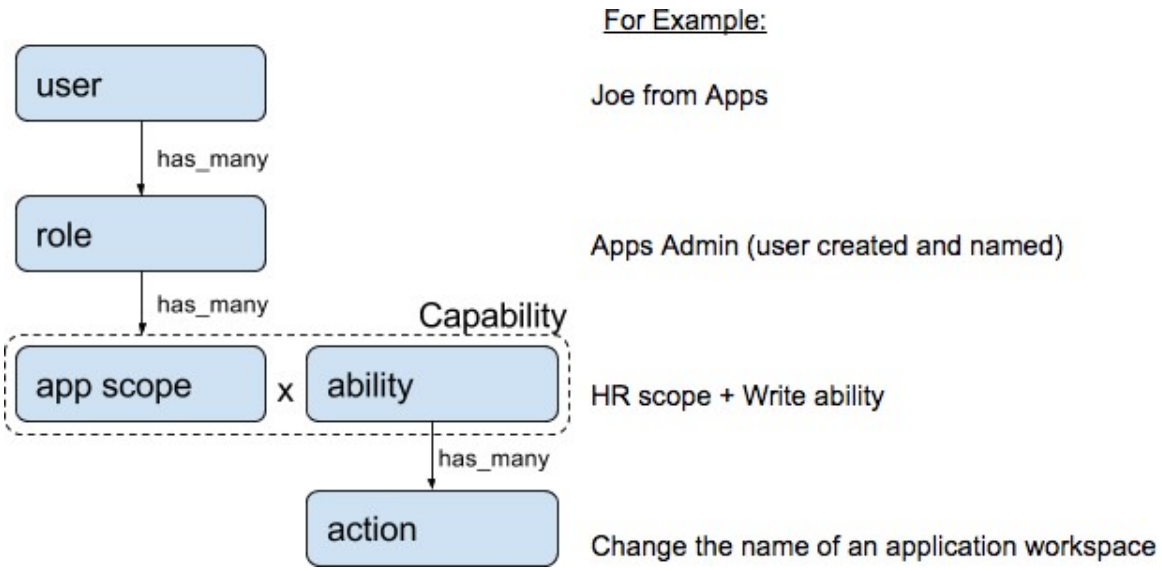
ログイン時に二要素認証コードを入力する必要はなくなりました。

ロール

ロールベースアクセス コントロール（RBAC）モデルを使用して、機能とデータへのアクセスを制限できます。

- ユーザー：Cisco Secure Workload へのログインアクセス権を持つユーザー
- ロール：ユーザーが作成した、ユーザーに割り当てることが可能な一連の機能
- 機能：範囲 + 能力のペア
- 能力：アクションの集合
- アクション：「ワークスペース名の変更」などの低レベルのユーザーアクション

図 *590 :* ロールモデル



ユーザーは、任意の数のロールを持つことができます。ロールには、任意の数の機能を含めることができます。たとえば、「HR 検索エンジニア」ロールには、可視性とコンテキストを提供する「HR範囲での読み取り」機能と、このロールを割り当てられたエンジニアがアプリケーションに関連した特定の変更を実施できるようにする「HR 検索の実行」機能を含めることが可能です。

ロールには一連の機能が含まれ、[ユーザー（Users）]ページでユーザーに割り当てられます。ユーザーは、任意の数のロールを持つことができます。ロールには、任意の数の機能を含めることができます。

ロール	説明
エージェントインストーラ	インストール、モニター、アップグレード、変換を含むエージェントのライフサイクルを管理する機能を提供しますが、エージェントを削除したりエージェント設定プロファイルにアクセスしたりすることは**できません**。
カスタマー サポート	テクニカルサポートまたはアドバンスドサービスの場合。クラスタメンテナンス機能へのアクセスが可能です。サイト管理者と同じアクセス権が与えられますが、ユーザーを変更することは**できません**。
サイト管理者	ユーザー、エージェントなどを管理する能力を提供します。すべての機能とデータを表示および編集できます。少なくとも 1 人のサイト管理者が必要です。
グローバルアプリケーション適用	すべての範囲での適用能力を提供します。

ロール	説明
グローバルアプリケーション管理	すべての範囲での実行能力を提供します。
グローバル読み取り専用	すべての範囲での読み取り能力を提供します。

ロール	説明
エージェントインストーラ	インストール、モニター、アップグレード、変換を含むエージェントのライフサイクルを管理する機能を提供しますが、エージェントを削除したりエージェント設定プロファイルにアクセスしたりすることは**できません**。
カスタマー サポート	テクニカルサポートまたはアドバンスドサービスの場合。クラスタメンテナンス機能へのアクセスが可能です。サイト管理者と同じアクセス権が与えられますが、ユーザーを変更することは**できません**。
カスタマーサポート読み取り専用	テクニカルサポートまたはアドバンスドサービスの場合。クラスタメンテナンス機能へのアクセスが可能です。サイト管理者と同じアクセス権が与えられますが、ユーザーを変更することは**できません**。
サイト管理者	ユーザー、エージェントなどを管理する能力を提供します。すべての機能とデータを表示および編集できます。少なくとも 1 人のサイト管理者が必要です。
グローバルアプリケーション適用	すべての範囲での適用能力を提供します。
グローバルアプリケーション管理	すべての範囲での実行能力を提供します。
グローバル読み取り専用	すべての範囲での読み取り能力を提供します。

能力と機能

ロールは、範囲と能力を含む機能で構成されます。これらは、許可されるアクション、およびそれらが適用されるデータのセットを定義します。たとえば、（HR、読み取り） 機能は、「HR 範囲における読み取り能力」として解釈される必要があります。この機能により、HR 範囲とそのすべての子にアクセスできます。

能力	説明
読み取り	フロー、アプリケーション、インベントリフィルタを含むすべてのデータを読み取ります。

能力	説明
書き込み	アプリケーションとインベントリフィルタに変更を加えます。
実行	ポリシーの自動検出を実行し、分析のためにポリシーを公開します。
適用	指定された範囲に関連付けられたアプリケーション ワークスペースで定義されたポリシーを適用します。
オーナー	アプリケーション ワークスペースをセカンダリからプライマリに切り替えるために必要です。ユーザー アプリケーション セッションの管理、データタップの追加、視覚化データソースの作成などのデータタップ管理機能へのアクセス。

重要 能力は継承されます。たとえば、実行能力では、読み取り、書き込み、および実行アクションがすべて許可されます。

重要 能力は、範囲および範囲のすべての子に適用されます。

ロール別のメニューアクセス

ユーザーが表示および使用できるメニューは、ユーザーに割り当てられたロールによって異なります。

表 *40 : [*概要（*Overview*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
概要	概要	対応	対応	対応	対応	対応	非対応

表 *41 : [*概要（*Overview*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
概要	概要	対応	対応	対応	対応	対応	対応	非対応

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
レポート	[概要（Overview）]	対応	対応	対応	対応	対応	対応	非対応

表 *42 : [*整理（*Organize*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[整理（Organize）]	[範囲とインベントリ（Scopes and Inventory）]	対応	対応	対応	対応	対応	非対応
[整理（Organize）]	アップロードラベルの使用（Use Uploaded Labels）	対応	対応	非対応	非対応	非対応	非対応
[整理（Organize）]	[インベントリフィルタ（Inventory Filters）]	対応	対応	対応	対応	対応	非対応

表 *43 : [*整理（*Organize*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[整理（Ogmie）]	[範囲とインベントリ（Scopes and Inventory）]	対応	対応	対応	対応	対応	対応	非対応
[整理（Ogmie）]	Label Management	対応	対応	対応	対応	対応	対応	非対応
[整理（Ogmie）]	[インベントリフィルタ（Inventory Filters）]	対応	対応	対応	対応	対応	対応	非対応

表 *44 : [*防御（*Defend*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[防御（Defend）]	[セグメンテーション（Sgmentation）]	対応	対応	対応	対応	非対応	非対応
[防御（Defend）]	[適用ステータス（Enforcement Status）]	対応	対応	非対応	非対応	非対応	非対応
[防御（Defend）]	[ポリシーテンプレート（Policy Templates）]	対応	対応	非対応	非対応	非対応	非対応

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[防御（Defend）]	[フォレンジックルール（Forensic Rules）]	対応	対応	非対応	非対応	非対応	非対応

表 *45 : [*防御（***Defend***）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[防御（Defend）]	[セグメンテーション（Segmentation）]	対応	対応	対応	対応	対応	対応	非対応
[防御（Defend）]	[適用ステータス（Enforcement Status）]	対応	対応	対応	対応	対応	対応	非対応
[防御（Defend）]	[ポリシーテンプレート（Policy Templates）]	対応	対応	対応	対応	対応	対応	非対応
[防御（Defend）]	[フォレンジックルール（Forensic Rules）]	対応	対応	対応	対応	対応	対応	非対応

表 *46 : [*調査（*investigate*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[調査（Investigate）]	[トラフィック（Traffic）]	対応	対応	対応	対応	対応	非対応
[調査（Investigate）]	[アラート（Alerts）]	対応	対応	対応	対応	対応	非対応
[調査（Investigate）]	[脆弱性（Vulnerabilities）]	対応	対応	対応	対応	対応	非対応
[調査（Investigate）]	[フォレンジック（Forensics）]	対応	対応	対応	対応	対応	非対応
[調査（Investigate）]	[近隣（Neighborhood）]	対応	対応	対応	対応	対応	非対応

表 *47 : [*調査（*investigate*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[調査（Investigate）]	[トラフィック（Traffic）]	対応	対応	対応	対応	対応	対応	非対応
	[アラート（Alerts）]	対応	対応	対応	対応	対応	対応	非対応
	[脆弱性（Vulnerabilities）]	対応	対応	対応	対応	対応	対応	非対応
	[フォレンジック（Forensics）]	対応	対応	対応	対応	対応	対応	非対応

表 *48 : [*管理（*Manage*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[管理（Manage）]	[エージェント（Agents）]	対応	対応	非対応	非対応	非対応	対応
[管理（Manage）]	[アラート設定（Alerts Configs）]	対応	対応	対応	対応	対応	非対応
[管理（Manage）]	[変更ログ（Change Logs）]	対応	非対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[コネクタ（Connectors）]	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[外部オーケストレータ（External Orchestrators）]	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[セキュアコネクタ（Secure Connector）]	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[仮想アプライアンス（Virtual Appliances）]	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[ユーザー（Users）]	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	ロール	対応	対応	非対応	非対応	非対応	非対応

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[管理（Manage）]	[脅威インテリジェンス（Threat Intelligence）]	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[ライセンス（Licenses）]	対応	非対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[収集ルール（Collection Rules）]	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[セッション設定（Session Configuration）]	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[使用状況分析（Usage Analytics）]	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[データタップ管理（Data Tap Admin）]	対応	非対応	非対応	非対応	非対応	非対応

表 *49 : [*管理（*Manage*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[管理（Manage）]	[エージェント（Agents）]	対応	対応	対応	非対応	非対応	非対応	対応
[管理（Manage）]	[アラート設定（Alerts Configs）]	対応	対応	対応	対応	対応	対応	非対応
[管理（Manage）]	[変更ログ（Change Logs）]	対応	非対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[コネクタ（Connectors）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[外部オーケストレータ（External Orchestrators）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[セキュアコネクタ（Secure Connector）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[仮想アプライアンス（Virtual Appliances）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[ユーザー（Users）]	対応	対応	非対応	非対応	非対応	非対応	非対応

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[管理（Manage）]	ロール	対応	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[脅威インテリジェンス（Threat Intelligence）]	対応	対応	対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[ライセンス（Licenses）]	対応	非対応	非対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[収集ルール（Collection Rules）]	対応	対応	対応	対応	対応	対応	非対応
[管理（Manage）]	[セッション設定（Session Configuration）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[使用状況分析（Usage Analytics）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[管理（Manage）]	[データタップ管理（Data Tap Admin）]	対応	非対応	非対応	非対応	非対応	非対応	非対応

表 *50 : [*プラットフォーム（*Platform*）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[プラットフォーム（Platform）]	[テナント（Tenants）]	対応	対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[クラスタ設定（Cluster Configuration）]	対応	対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[アウトバウンドHTTP（Outbound HTTP）]	対応	対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	コレクタ	対応	対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[外部認証（External Authentication）]	対応	対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[SSL証明書（SSL Certificate）]	対応	対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[ログインページメッセージ（Login Page Message）]	対応	対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[連携（Federation）]	以下を参照	以下を参照	非対応	非対応	非対応	非対応

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[プラットフォーム（Platform）]	[データのバックアップ（Data Backup）]	以下を参照	以下を参照	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[データの復元（Data Restore）]	以下を参照	以下を参照	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[アップグレード/再起動/シャットダウン（Upgrade/Reboot/Shutdown）]	対応	対応	非対応	非対応	非対応	非対応

表 *51 : [*プラットフォーム（***Platform***）*]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[プラットフォーム（Platform）]	[テナント（Tenants）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[クラスタ設定（Cluster Configuration）]	対応	対応	非対応	非対応	非対応	非対応	非対応

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[プラットフォーム（Platform）]	[アウトバウンドHTTP（Outbound HTTP）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	コレクタ	対応	対応	非対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[外部認証（External Authentication）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[SSL証明書（SSL Certificate）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[ログインページメッセージ（Login Page Message）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[連携（Federation）]	以下を参照	以下を参照	非対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[データのバックアップ（Data Backup）]	以下を参照	以下を参照	非対応	非対応	非対応	非対応	非対応

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[プラットフォーム（Platform）]	[データの復元（Data Restore）]	以下を参照	以下を参照	非対応	非対応	非対応	非対応	非対応
[プラットフォーム（Platform）]	[アップグレード/再起動/シャットダウン（Upgrade/Reboot/Shutdown）]	対応	対応	非対応	非対応	非対応	非対応	非対応

（注）
- 連携が有効になっている場合、サイト管理者およびカスタマーサポートのロールで [連携（Federation）] オプションを使用できます。
- データのバックアップとデータの復元が有効になっている場合、サイト管理者とカスタマーサポートのロールで、[データのバックアップ（Data Backup）] と [データの復元（Data Restore）] オプションを使用できます。

表 *52 : [*トラブルシューティング（*Troubleshooting]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[トラブルシューティング（Troubleshoot）]	サービスのステータス	対応	対応	非対応	非対応	非対応	非対応

メニュー	オプション	サイト管理者	カスタマーサポート	アプリケーションのグローバル管理	グローバルに読み取り専用	アプリケーションのグローバル適用	エージェントインストーラ
[トラブルシューティング（Troubleshoot）]	[クラスタのステータス（Cluster Status）]	以下を参照	以下を参照	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshoot）]	[仮想マシン（Virtual Machine）]	対応	対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshoot）]	[スナップショット（Snapshots）]	対応	対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshoot）]	[メンテナンスエクスプローラ（Maintenance Explorer）]	対応	対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshoot）]	[救済（Resque）]	対応	対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshoot）]	[Hawkeye]（チャート）	対応	対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshoot）]	[Abyss]（パイプライン）	対応	対応	非対応	非対応	非対応	非対応

表 *53 : [*トラブルシューティング（*Troubleshooting]* メニュー

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[トラブルシューティング（Troubleshooting）]	サービスのステータス	対応	対応	対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshooting）]	[クラスタのステータス（Cluster Status）]	以下を参照	以下を参照	非対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshooting）]	[仮想マシン（Virtual Machine）]	対応	対応	対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshooting）]	[スナップショット（Snapshots）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshooting）]	[メンテナンスエクスプローラ（Maintenance Explorer）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshooting）]	[救済（Rescue）]	対応	対応	非対応	非対応	非対応	非対応	非対応
[トラブルシューティング（Troubleshooting）]	[Hawkeye]（チャート）	対応	対応	対応	非対応	非対応	非対応	非対応

メニュー	オプション	サイト管理者	カスタマーサポート	カスタマーサポート読み取り専用	アプリケーションのグローバル適用	アプリケーションのグローバル管理	グローバルに読み取り専用	エージェントインストーラ
[トラブルシューティング（Troubleshooting）]	[Abyss]（パイプライン）	対応	対応	対応	非対応	非対応	非対応	非対応

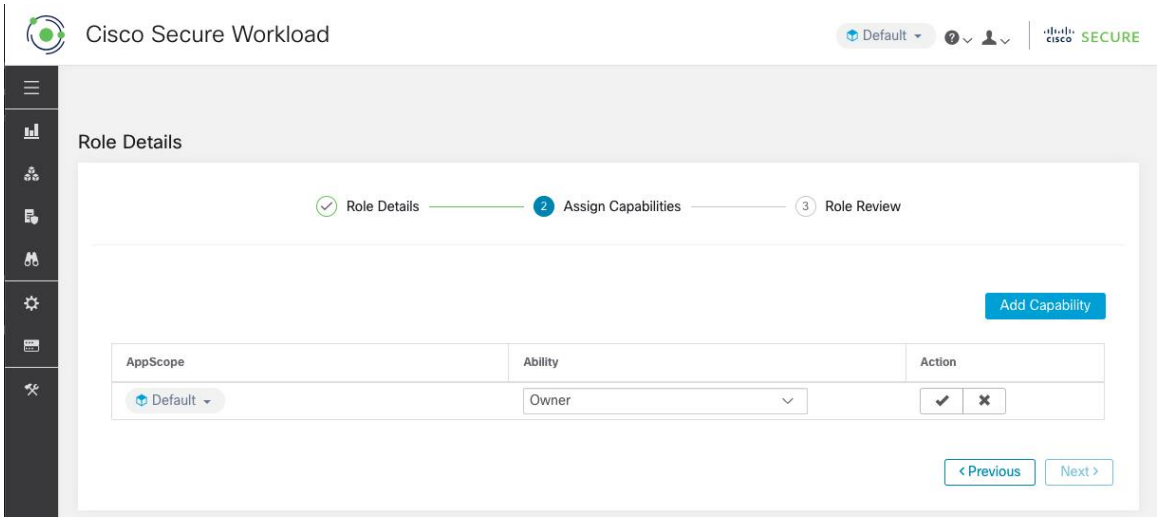
（注） [クラスタのステータス（Cluster Status）] オプションは、クラスタタイプが「物理」または「OCI」の場合に、サイト管理者およびカスタマーサポートのロールで使用できます。

新しいロールの作成

始める前に

[サイト管理者（Site Admin）] または [カスタマーサポート（Customer Support）] のユーザーロールが割り当てられている必要があります。

1. 左側のナビゲーションバーで、**[管理（Manage）]**>**[ロール（Roles）]** をクリックします。
2. [ロールの作成（Create Role）] ボタンをクリックします。[ロール（Roles）] パネルが表示されます。



[ロールの作成（Create Role）] ウィザードを使用したロールの作成は、3 つのステップから成るプロセスです。

手順

ステップ 1 a) 以下のフィールドに適切な値を入力します。

フィールド	説明
名前	ロールを識別するための名前。
[説明（Description）]	ロールに関するコンテキストを追加するための簡単な説明。

b) [次へ（Next）] ボタンをクリックして次のステップに移動するか、[ロールページに戻る（Back to Roles Page）] をクリックして [ロール（Roles）] ページに戻ります。

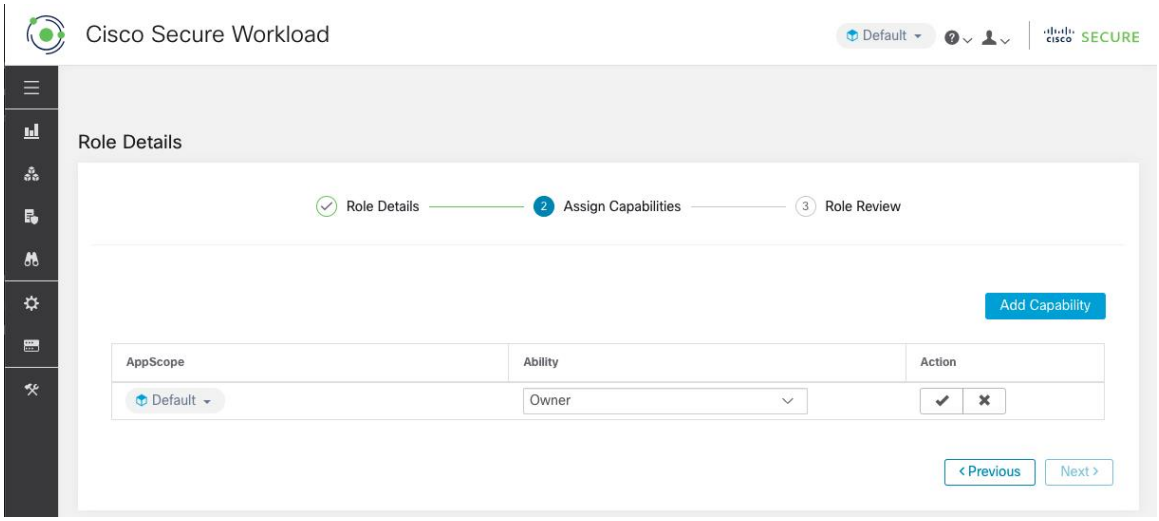
ステップ 2 a) [ケーパビリティの追加（Add Capability）] ボタンをクリックすると、一番上の行に作成フォームが表示されます。

b) 範囲と権限を選択します。

c) チェックマークボタンをクリックして新しいケーパビリティを作成するか、キャンセルボタンをクリックしてキャンセルします。

d) [次へ（Next）] ボタンをクリックしてロールの詳細を確認するか、[前へ（Previous）] をクリックして戻って編集します。

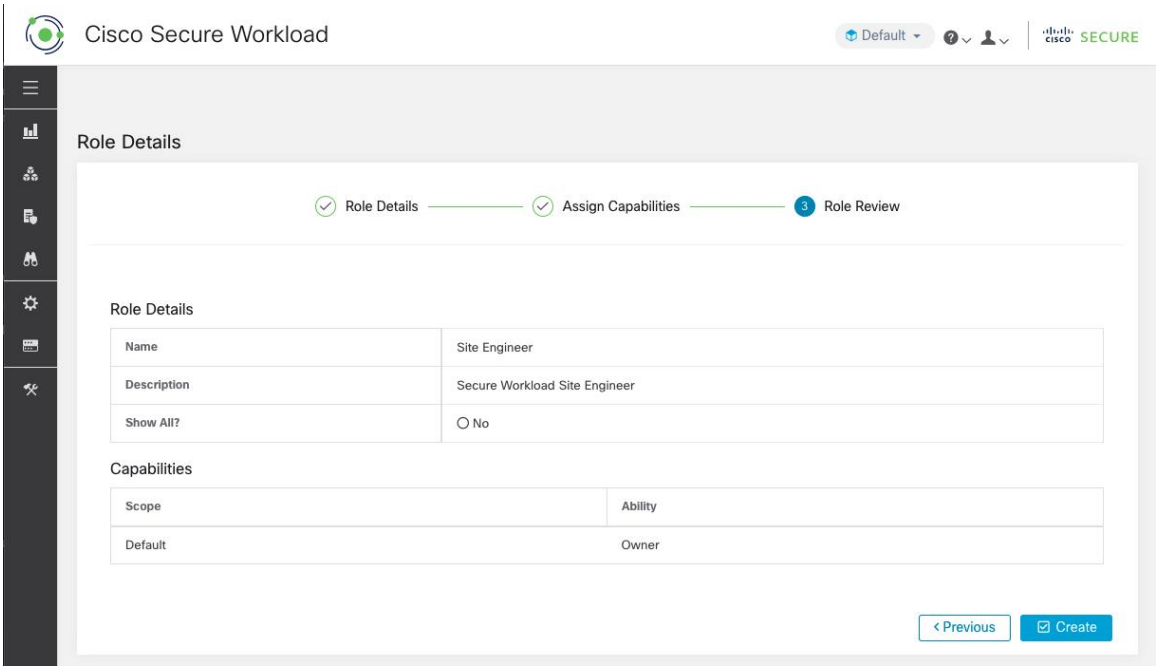
図 *591* : ケーパビリティの割り当て



ステップ 3 a) ロールの詳細とケーパビリティを確認します。

b) [作成（Create）] をクリックして、ロールを作成します。

図 *592* : ロールの確認



ロールの編集

このセクションでは、**サイト管理者**と**カスタマーサポートユーザー**がロールを編集する方法について説明します。

始める前に

サイト管理者またはカスタマーサポートユーザーである必要があります。

1. 左側のナビゲーションバーで、**[管理（Manage）]**>**[ロール（Roles）]**をクリックします。
2. 編集するロールの行で、右側の列にある [編集（Edit）] ボタンをクリックします。[ロール（Roles）] パネルが表示されます。

[ロールの編集（Edit Role）] ウィザードを使用したロールの編集は、3 つのステップから成るプロセスです。

手順

ステップ 1
a) 必要に応じて、名前や説明を更新します。
b) [次へ（Next）] ボタンをクリックして次のステップに移動するか、[ロールページに戻る（Back to Roles Page）] をクリックして [ロール（Roles）] ページに戻ります。

ステップ 2
a) 必要に応じてケーパビリティを削除します。削除するケーパビリティの行で、右側の列にある [削除（Delete）] アイコンをクリックします。
b) 追加するには、[ケーパビリティの追加（Add Capability）] ボタンをクリックして、一番上の行に作成フォームを表示します。
c) 範囲と権限を選択します。
d) [次へ（Next）] ボタンをクリックしてロールの詳細を確認するか、[前へ（Previous）] をクリックして戻って編集します。

ステップ 3
a) ロールの詳細とケーパビリティを確認します。
b) [更新（Update）] をクリックしてロールを作成するか、[前へ（Previous）] をクリックして編集します。ロールの詳細とケーパビリティの割り当てへの変更は、[更新（Update）] をクリックした後に保存されます。

（注） ケーパビリティは編集できません。削除して作成し直す必要があります。

変更ログ：役割

ルート範囲で SCOPE_OWNER 機能を持つ [サイト管理者（Site Admins）] およびユーザーは、以下に示す [アクション（Action）] 列のアイコンをクリックして、各ロールのログの変更を表示できます。

図 *593 :* ログの変更



該当するユーザーは、テーブルの下にある [削除されたロールを表示（View Deleted Roles）] リンクをクリックして、削除されたロールのリストを表示することもできます。

ログの変更の詳細については、「ログの変更」を参照してください。ルート範囲の所有者は、その範囲に属するエンティティにおけるログの変更エントリの表示に制限されます。

スコープ

（注） [範囲（Scopes）] ページは [インベントリ検索（Inventory Search）] と統合されました。以下のリンクのヘルプについては、[範囲とインベントリ（Scopes and Inventory）] ページを参照してください。

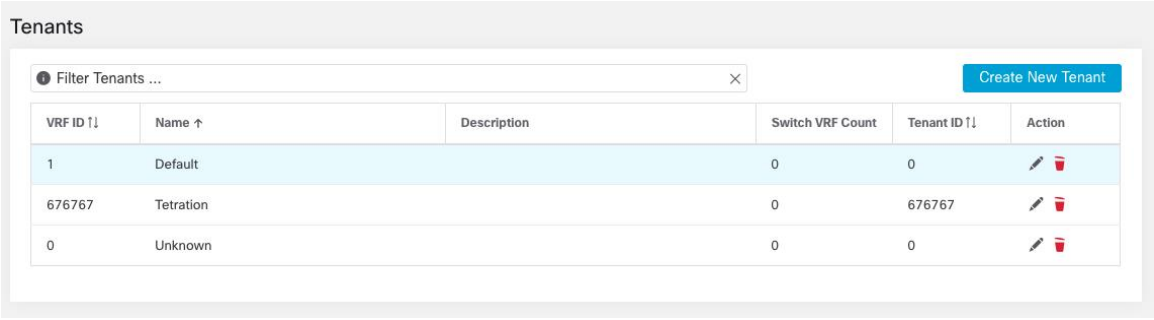
範囲とインベントリ

テナント

サイト管理者および**カスタマーサポートユーザー**は、左側にあるナビゲーションバーの**[プラットフォーム（Platform）]** > **[テナント（Tenants）]** メニューの下にある [テナント（Tenants）] ページにアクセスできます。このページには、現在構成されているすべてのテナントと VRF が表示されます。システムには、1 つ以上のテナントと VRF が事前設定されています。テナントの追加、編集、削除が可能です。

（注） これらの値は、クラスタ出力の結果に影響します。システムへの影響を理解するために、これらの値を変更する前に Cisco TAC に相談することをお勧めします。

図 *594* : テナント ページ



テナントの追加

始める前に

[サイト管理者（Site Admin）] または [カスタマーサポート（Customer Support）] ユーザーである必要があります。

手順

ステップ 1 左側のナビゲーションバーで、**[プラットフォーム（Platform）]** > **[テナント（Tenants）]** をクリックします。

ステップ 2 [新しいテナントの作成（Create New Tenant）] をクリックします。

ステップ 3 以下のフィールドに適切な値を入力します。

フィールド	説明
名前	テナントの名前を入力します。

フィールド	説明
説明	（オプション）説明フィールドには、テナントに関する追加情報が含まれています。

ステップ 4 [作成（Create）] をクリックします。

テナントの編集

始める前に

[サイト管理者（Site Admin）] または [カスタマーサポート（Customer Support）] ユーザーである必要があります。

手順

ステップ 1 左側のナビゲーションバーで、**[プラットフォーム（Platform）]** > **[テナント（Tenants）]** をクリックします。

ステップ 2 編集するテナントを見つけて、右側の列にある**鉛筆**アイコンをクリックします。

フィールド	説明
名前	テナントの名前を更新します。
説明	（オプション）テナントに関する追加情報が含まれている説明フィールドを更新します。
VRF ID	この特定のテナント/VRF の ID を表示します。
変更ログ	変更ログアイコンをクリックすると、テナント/VRF のすべての変更ログを示す新しいページに移動します。

ステップ 3 [更新（Update）]をクリックします。

ユーザ（Users）

サイト管理者とルート範囲の所有者は、ウィンドウの左側にあるナビゲーションバーの [管理（Manage）] メニューから [ユーザー（Users）] ページにアクセスできます。

このページには、すべてのサービス プロバイダー ユーザーと、ページヘッダーで選択した範囲に関連付けられているユーザーが表示されます。

マルチテナント機能

マルチテナント機能をサポートするために、ユーザーをルート範囲に割り当てることができます。これらのユーザーは、ルート範囲で「所有者」権限を持つユーザーによって管理され、同じ範囲に関連付けられたロールのみを割り当てることができます。

範囲のないユーザーは「サービスプロバイダー」と呼ばれ、これらのユーザーにはルート範囲全体でアクションを実行できる任意のロールを割り当てることができます。

新しいユーザーアカウントの追加

このセクションでは、**サイト管理者**およびルート範囲で**SCOPE_OWNER**機能を持つユーザーが新しいユーザーアカウントを追加する方法について説明します。

マルチテナント機能のためにユーザーにある範囲が割り当てられている場合、同じ範囲に割り当てられたロールのみを選択できます。

（注） このページは、ページヘッダーで選択された範囲設定によってフィルタリングされます。

始める前に

1. 左側のナビゲーションバーで、**[管理（Manage）]** > **[ユーザー（Users）]** をクリックします。
2. 該当する場合は、ページの右上から適切なルート範囲を選択します。
3. [新しいユーザーの追加（Add New User）] ボタンをクリックします。[ユーザー（Users）] ウィザードが表示されます。

ユーザーの作成は 3 段階のプロセスです。

手順

ステップ 1 1. 以下のフィールドに適切な値を入力します。

フィールド	説明
E メール（Email）	新しいユーザーの電子メールアドレスを入力します。大文字と小文字は区別されません。メールに文字が含まれている場合は、小文字のバージョンを使用します。
[名（First Name）]	新しいユーザーの名を入力します。
[姓（Last Name）]	新しいユーザーの姓を入力します。
スコープ	マルチテナンシーのためにユーザーに割り当てられたルート範囲。

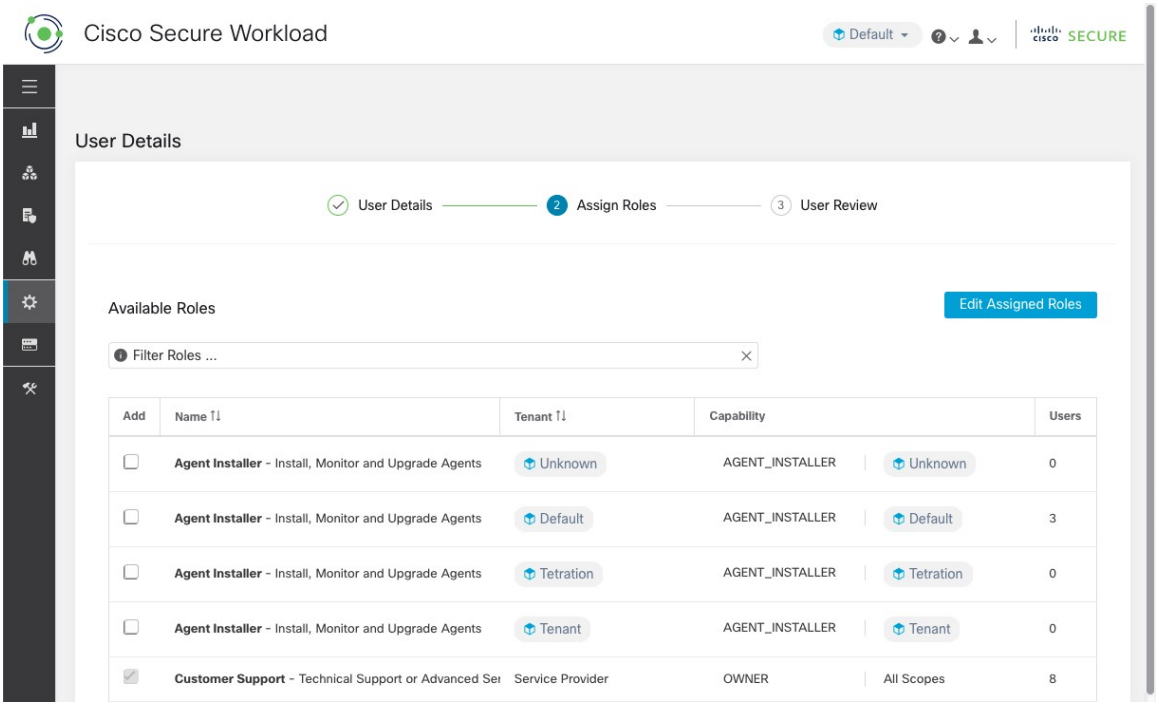
必要に応じて、SSH公開キーを今すぐインポートすることも、後でインポートすることもできます。

2. [次へ（Next）]ボタンをクリックして次のステップに移動するか、[ユーザーリストに戻る（Back to Users List）]をクリックして[ユーザー（Users）]ページに戻ります。

ステップ2 このビューでは、[ロールの追加（Add Roles）]、[ロールの削除（Delete Roles）]、または[ロールの選択（Select Roles）]を実行できます。

1. [ロールの追加（Add Roles）]をクリックして、ロールを割り当てます。

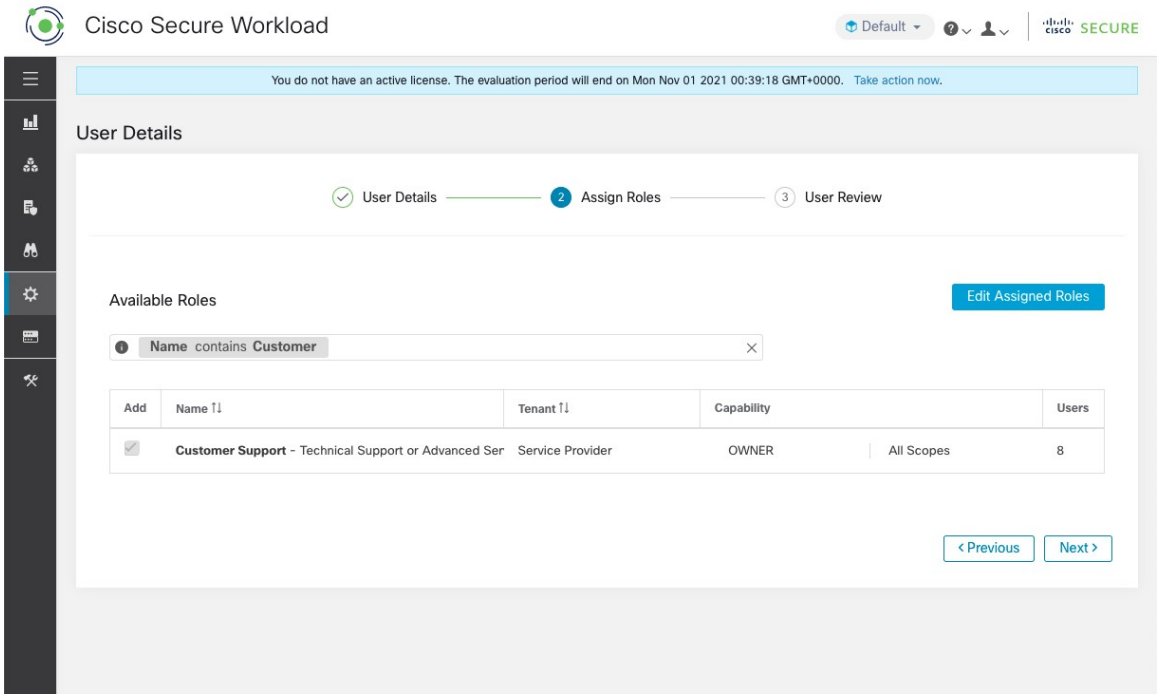
図 *595 :* 使用可能な役割（*Available Roles*）



2. [割り当てられたロールの編集（Edit Assigned Roles）]をクリックしてロールを削除します。

3. [名前（Name）]と[テナント（Tenant）]でロールをフィルタ処理します。

図 *596* : ロールをフィルタ処理



4. [次へ（Next）] ボタンをクリックしてユーザーの詳細とロールの割り当てを確認するか、[前へ（Previous）] ボタンをクリックして戻って詳細を編集します。

ステップ 3 設定内容を確認して [作成（Create）] をクリックします。

外部認証が有効になっている場合、認証の詳細が表示されます。

（注） ユーザーを作成すると、そのユーザーはパスワードを設定するための電子メールを受け取ります。

ユーザー アカウントの編集

始める前に

サイト管理者または**ルート範囲所有者**ユーザーの権限が必要です。

（注） このページは、ページヘッダーで選択された範囲設定に従ってフィルタリングされます。

1. 左側のナビゲーションバーで、**[管理（Manage）]** > **[ユーザー（Users）]** をクリックします。

2. 該当する場合は、ページの右上から適切なルート範囲を選択します。
3. 編集するアカウントの行で、右側の列にある [編集（Edit）] ボタンをクリックします。[ユーザー（Users）] ウィザードが表示されます。

ウィザードを使用したユーザーの編集は、3 つのステップから成るプロセスです。

手順

ステップ 1

1. 必要に応じて、次のフィールドを更新します。

フィールド	説明
E メール（Email）	新規ユーザーの電子メールアドレスを更新します。
[名（First Name）]	新規ユーザーの名を更新します。
[姓（Last Name）]	新規ユーザーの姓を更新します。
スコープ	マルチテナンシーのためにユーザーに割り当てられたルート範囲。（サイト管理者が利用可能）

2. [次へ（Next）] ボタンをクリックして、[ロールの割り当て（Role Assignment）] に進みます。

ステップ 2

1. このビューでは、割り当てられたロールを削除できます。
2. [ロールの追加（Add Roles）] をクリックして、新しいロールを割り当てます。
3. [次へ（Next）] ボタンをクリックしてユーザーの詳細とロールの割り当てを確認するか、[前へ（Previous）] ボタンをクリックして戻って詳細を編集します。

ステップ 3

1. ユーザーの詳細とロールの割り当てを確認します。
2. [更新（Update）]をクリックしてユーザーを更新するか、[前へ（Previous）]をクリックして戻ってロールを編集します。ユーザーの詳細とロールの割り当てへの変更が保存されます。

外部認証が有効になっている場合、認証の詳細が表示されます。

ユーザーアカウントの非アクティブ化

（注） 変更ログ監査の一貫性を維持するために、ユーザーは非アクティブ化できますが、データベースからは削除されません。

始める前に

サイト管理者または**ルート範囲所有者**ユーザーの権限が必要です。

（注） このページは、ページヘッダーで選択された範囲設定に従ってフィルタリングされます。

手順

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）] > [ユーザー（Users）]** をクリックします。

ステップ 2 該当する場合は、ページの右上から適切なルート範囲を選択します。

ステップ 3 非アクティブ化するアカウントの行で、右側の列にある [非アクティブ化（Deactivate）] ボタンをクリックします。

非アクティブ化されたユーザーを表示するには、[削除されたユーザーを非表示（Hide Deleted Users）] ボタンを切り替えます。

ユーザーアカウントの再アクティブ化

ユーザーが非アクティブ化されている場合は、そのユーザーを再アクティブ化できます。

始める前に

サイト管理者または**ルート範囲所有者**ユーザーの権限が必要です。

（注） このページは、ページヘッダーで選択された範囲設定に従ってフィルタリングされます。

手順

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）] > [ユーザー（Users）]** をクリックします。

ステップ 2 該当する場合は、ページの右上から適切なルート範囲を選択します。

ステップ 3 [削除されたユーザーを非表示（Hide Deleted Users）] ボタンを切り替えると、非アクティブ化されたユーザーを含むすべてのユーザーが表示されます。

ステップ 4 再アクティブ化する非アクティブ化されたアカウントの行で、右側の列にある [復元（Restore）] ボタンをクリックします。

SSH 公開キーのインポート

コレクタ IP アドレスの 1 つを介した **ta_guest** ユーザーとしての SSH アクセスを有効にするために、SSH 公開キーをユーザーごとにインポートできます。このメニューは、**サイト管理者**およびルート範囲で `SCOPE_OWNER` 機能を持つユーザーのみが使用できます。SSH 公開キーは、7 日後に自動的に期限切れになります。

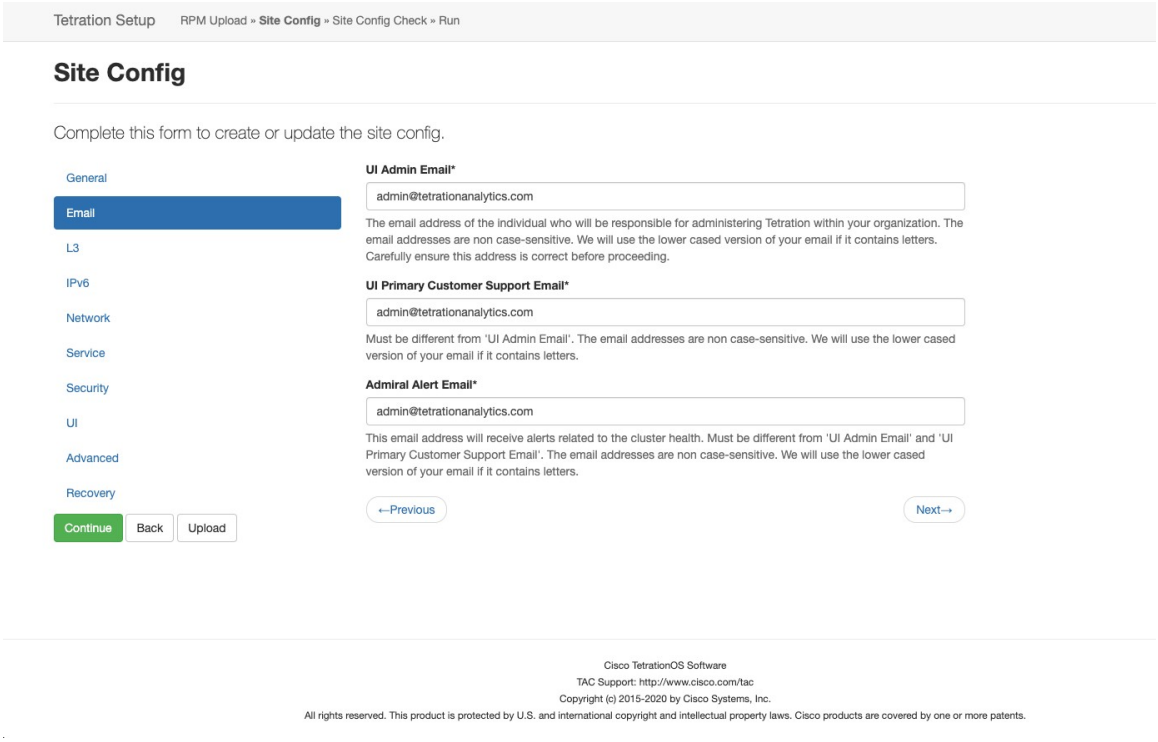
Secure Workload セットアップでのサイト設定

このセクションでは、[サイト管理者（Site Admins）] が Secure Workload のセットアッププロセス中にサイトをセットアップする方法について説明します。

フィールド	説明
[UI管理者の電子メール（UI Admin Email）]	組織内で Secure Workload の管理を担当する個人の電子メールアドレス。
[UIプライマリカスタマーサポートの電子メール（UI Primary Customer Support Email）]	プライマリサポートの電子メールアドレス。UI 管理者の電子メールとは別にする必要があります。
[アドミラルアラート電子メール（Admiral Alert Email）]	この電子メールアドレスは、クラスタ正常性に関連するアラートを受信します。UI管理者の電子メールおよびUIプライマリカスタマサポートの電子メールとは別にする必要があります。

電子メールアドレスでは大文字と小文字が区別されません。電子メールに文字が含まれている場合は、小文字のバージョンを使用します。

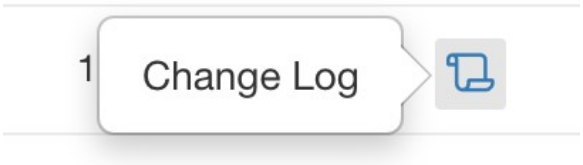
図 *597 : UI* 管理者、プライマリ カスタマー サポート、アドミラル管理者アラート メールの構成



ログの変更：ユーザー

ルート範囲で SCOPE_OWNER 機能を持つ [サイト管理者（Site Admins）] およびユーザーは、以下に示す [アクション（Actions）] アイコンをクリックして、各ユーザーのログの変更を表示できます。

図 *598 :* ログの変更



ログの変更の詳細については、「ログの変更」を参照してください。ルート範囲の所有者は、その範囲に属するエンティティにおけるログの変更エントリの表示に制限されます。



第 18 章

OpenAPI

OpenAPI は、Secure Workload 機能用の REST API を提供します。

- OpenAPI 認証 （1056 ページ）
- ワークスペースとセキュリティポリシー （1058 ページ）
- スコープ （1121 ページ）
- アラート （1127 ページ）
- ロール （1131 ページ）
- ユーザ（Users） （1135 ページ）
- インベントリ フィルタ （1140 ページ）
- フロー検索 （1144 ページ）
- インベントリ （1152 ページ）
- ワークロード （1157 ページ）
- デフォルトのポリシー生成設定 （1167 ページ）
- フォレンジックインテント （1170 ページ）
- フォレンジックインテント順序 （1172 ページ）
- フォレンジックプロファイル （1174 ページ）
- フォレンジックルール （1176 ページ）
- プラットフォーム設定 （1179 ページ）
- 施行 （1183 ページ）
- クライアントサーバー構成 （1192 ページ）
- ソフトウェアエージェント （1196 ページ）
- Cisco Secure Workload ソフトウェアのダウンロード （1204 ページ）
- Cisco Secure Workload エージェントのアップグレード （1207 ページ）
- ユーザーがアップロードしたファイルハッシュ （1208 ページ）
- ユーザー定義ラベル （1210 ページ）
- Virtual Routing and Forwarding（VRF） （1221 ページ）
- オーケストレーション （1225 ページ）
- オーケストレータのゴールデンルール （1232 ページ）
- FMC オーケストレータドメイン （1233 ページ）
- RBAC（役割ベースのアクセス制御）に関する考慮事項 （1236 ページ）

• 高可用性とフェールオーバーに関する考慮事項 （1236 ページ）
• Kubernetes RBAC リソースに関する考慮事項 （1236 ページ）
• Service Health （1238 ページ）
• Secure Connector （1238 ページ）
• 外部オーケストレータのポリシー適用ステータス （1240 ページ）
• 管理対象データタップとデータシンクの証明書のダウンロード （1241 ページ）
• 変更ログ （1243 ページ）
• ルーティング不可能なエンドポイント （1245 ページ）
• 外部アプライアンスおよびコネクタの設定およびコマンドのスキーマ （1248 ページ）

OpenAPI 認証

OpenAPI はダイジェスト ベースの認証方式を使用します。ワークフローは次のようになります。

1. Secure Workload UI ダッシュボードにログインします。
2. 目的の機能を使用して API キーと API 秘密を生成します。
3. JSON 形式で REST 要求を送信するには、Secure Workload API SDK を使用します。
4. Python SDK を使用するには、`pip install tetpyclient` を使用して SDK をインストールします。
5. Python sdk をインストールしたら、RestClient をインスタンス化するためのいくつかの定型コードを次に示します。

```
from tetpyclient import RestClient

API_ENDPOINT="https://<UI_VIP_OR_DNS_FOR_TETRATION_DASHBOARD>"

# ``verify`` is an optional param to disable SSL server authentication.
# By default, cluster dashboard IP uses self signed cert after
# deployment. Hence, ``verify=False`` might be used to disable server
# authentication in SSL for API clients. If users upload their own
# certificate to cluster (from ``Platform > SSL Certificate``)
# which is signed by their enterprise CA, then server side authentication
# should be enabled; in such scenarios, in the code below, verify=False
# should be replaced with verify="path-to-CA-file"
# credentials.json looks like:
# {
#   "api_key": "<hex string>",
#   "api_secret": "<hex string>"
# }

restclient = RestClient(API_ENDPOINT,
                credentials_file='<path_to_credentials_file>/credentials.json',
                verify=False)
# followed by API calls, for example API to retrieve list of agents.
# API can be passed /openapi/v1/sensors or just /sensors.
resp = restclient.get('/sensors')
```

API キーと秘密の生成

手順

ステップ 1 Secure WorkloadのWebインターフェイスで、ウィンドウの右上隅にある人のアイコンをクリックし、[APIキー（API Keys）] を選択します。

ステップ 2 [API キーの作成（Create API Key）] をクリックします。

ステップ 3 キーと秘密に必要な機能を指定します。ユーザーは、API キー＋秘密のペアの使用を予定している機能の制限設定を選択する必要があります。ユーザーが使用できるAPI機能は、ユーザーのロールによって異なります。たとえば、サイト管理者ユーザーは、ソフトウェアエージェントを管理するためのキーを生成できますが、この機能は非サイト管理者ユーザーには使用できません。

API 機能には次のものがあります。

- SWエージェント管理（`sensor_management`）：SWエージェントのステータスを設定およびモニタリングできます
- Cisco Secure Workload ソフトウェアのダウンロード（`software_download`）：Secure Workloadエージェント/仮想アプライアンスのソフトウェアパッケージをダウンロードできます
- フローとインベントリ検索（`flow_inventory_query`）：Secure Workloadクラスタ内のフローとインベントリ項目を照会できます
- ユーザー、ロール、範囲の管理（`user_role_scope_management`）：ユーザー、ロール、範囲を読み取り/追加/変更/削除できます
- ユーザーデータのアップロード（`user_data_upload`）：ユーザーがフローとインベントリ項目に注釈を付けるためにデータをアップロードしたり、正当または不当なファイルハッシュをアップロードしたりできます
- ワークスペースとポリシー管理（`app_policy_management`）：ワークスペース（「アプリケーション」）を管理し、ポリシーを適用できます
- 外部システム統合: vCenter、Kubernetes などの外部システムと統合できます
- Cisco Secure Workload アプライアンス管理：Secure Workload クラスタを管理できます（サイト管理ユーザーのみが利用可能）

ステップ 4 [作成（Create）] をクリックします。

ステップ 5 キーと秘密をコピーして貼り付け、安全な場所に保存します。または、API クレデンシャルファイルをダウンロードします。

（注） LDAP による外部認証や LDAP 認証が有効になっている場合、ユーザーセッションが終了するとグループの LDAP メンバーから派生した Secure Workload ロールが再評価されるため、API キーを介した OpenAPI へのアクセスはシームレスに機能しなくなります。したがって、中断のない OpenAPI アクセスを保証するために、API キーを持つすべてのユーザーが [ユーザー詳細の編集（Edit User Details）] フローで [ローカル認証を使用（Use Local Authentication）] オプションを有効にすることを推奨します。

ワークスペースとセキュリティポリシー

次のページでは、ポリシーを管理するための OpenAPI エンドポイントについて説明します。

ワークスペース

ワークスペース（以前の呼称は「アプリケーションワークスペース」または「アプリケーション」）は、特定の範囲のポリシーを定義、分析、および適用するためのコンテナです。動作の詳細については、『ワークスペースを使用したポリシーの管理』のマニュアルを参照してください。この一連の API には、API キーに関連付けられている `app_policy_management` 機能が必要です。

ワークスペースオブジェクト

ワークスペース（「アプリケーション」）の JSON オブジェクトは、API エンドポイントに応じて、単一のオブジェクトまたはオブジェクトの配列として返されます。オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	ワークスペースの一意の識別子。
name	string	ユーザーが指定したワークスペース名。
説明	string	ユーザーが指定したワークスペースの説明。
app_scope_id	string	ワークスペースが関連付けられている範囲の ID。
author	string	ワークスペースを作成したユーザーの姓と名。
primary	boolean	ワークスペースがその範囲に対してプライマリであるかどうかを示します。

属性	タイプ	説明
alternate_query_mode	boolean	ワークスペースに「ダイナミックモード」が使用されているかどうかを示します。ダイナミックモードでは、自動ポリシー検出の実行により、クラスタごとに1つ以上の候補クエリが作成されます。デフォルト値は true です。
created_At	integer	ワークスペースが作成されたときの Unix タイムスタンプ。
latest_adm_version	integer	ワークスペースの最新の adm (v*) バージョン。
analysis_enabled	boolean	ワークスペースで分析が有効になっているかどうかを示します。
analyzed_version	integer	ワークスペースの分析された p* バージョン。
enforcement_enabled	boolean	ワークスペースで適用が有効になっているかどうかを示します。
enforced_version	integer	ワークスペースの適用された p* バージョン。

アプリケーションの一覧表示

このエンドポイントは、ワークスペース（「アプリケーション」）の配列を返します。

```
GET /openapi/v1/applications
```

表 *54 :* パラメータ

名前	タイプ	説明
app_scope_id	string	特定のアプリケーション範囲に関連付けられたワークスペースを照合します。
exact_name	string	ワークスペースを提供された値と正確に照合します。

応答オブジェクト：ワークスペースオブジェクトの配列を返します。

サンプル python コード

```
restclient.get('/applications')
```

単一のワークスペースの取得

このエンドポイントは、要求されたワークスペース（「アプリケーション」）を 1 つの JSON オブジェクトとして返します。

```
GET /openapi/v1/applications/{application_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

応答オブジェクト：指定された ID のワークスペースオブジェクトを返します。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
restclient.get('/applications/%s' % application_id)
```

ワークスペースの作成

このエンドポイントは、ワークスペース（「アプリケーション」）を作成します。クラスタとポリシーの定義を含む JSON 本文をポストすることによって、ポリシーを定義することができます。

（注） 同じ範囲にプライマリワークスペースが存在し、新しいポリシーが提供されている場合、ポリシーは新しいバージョンとして既存のワークスペースに追加されます。

```
POST /openapi/v1/applications
```

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
app_scope_id	string	ワークスペースに割り当てる範囲 ID。
name	string	（オプション）ワークスペースの名前。
説明	string	（オプション）ワークスペースの説明。

名前	タイプ	説明
alternate_query_mode	boolean	（オプション）ワークスペースに「ダイナミックモード」が使用されているかどうかを示します。ダイナミックモードでは、自動ポリシー検出の実行により、クラスタごとに1つ以上の候補クエリが作成されます。デフォルト値は true です。
strict_validation	boolean	（オプション）アップロードされたデータに不明なキー/属性がある場合、エラーを返します。スペルミスされたキーを検出するのに役立ちます。デフォルト値は false です。
プライマリ	string	（オプション）このワークスペースを関連付けられた範囲のプライマリにする必要がある場合は、「true」に設定します。**デフォルトは true です。**

ワークスペース内で作成されるポリシーを記述する、追加のオプションパラメータを含めることもできます。



（注） このスキームは、UI と**詳細**エンドポイントからのエクスポート時に返されるものに対応しています。

名前	タイプ	説明
クラスタ	クラスタの配列	ポリシーを定義するために使用されるノードのグループ。
inventory_filters	インベントリ フィルタの配列	データセンター アセットをフィルタリングします。
absolute_policies	ポリシーの配列	絶対ランクで作成される順序ポリシー。
default_policies	ポリシーの配列	デフォルト ランクで作成される順序ポリシー。
catch_all_action	string	「ALLOW」または「DENY」

クラスタ オブジェクト属性:

名前	タイプ	説明
id	string	ポリシーで使用される固有識別子。
name	string	クラスタの表示名。
説明	string	クラスタの説明。
nodes	ノードの配列	このクラスタに属するノードまたはエンドポイント 。
consistent_uuid	string	特定のワークスペースに対して一意である必要があります。自動ポリシー検出の実行後、類似/同じクラスタは次のバージョンでも consistent_uuid を維持します。

ノード オブジェクトの属性:

名前	タイプ	説明
ip	string	ノードの IP またはサブネット。例：10.0.0.0/8 または 1.2.3.4
name	string	ノードの表示名。

インベントリ フィルタ オブジェクトの属性:

名前	タイプ	説明
id	string	ポリシーで使用される固有識別子。
name	string	クラスタの表示名。
query	object	インベントリ フィルタ クエリの JSON オブジェクト表現。

ポリシー オブジェクトの属性:

名前	タイプ	説明
consumer_filter_id	string	クラスタの ID、ユーザー インベントリ フィルタ、またはアプリ範囲。
provider_filter_id	string	クラスタの ID、ユーザー インベントリ フィルタ、またはアプリ範囲。

名前	タイプ	説明
action	string	「ALLOW」または「DENY」
l4_params	l4params の配列	許可されたポートとプロトコルのリスト。

L4Params オブジェクト属性：

名前	タイプ	説明
proto	integer	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
port	array	ポートの包含範囲。例：[80, 80] または [5000, 6000]。
approved	boolean	（オプション）ポリシーが承認されているかどうかを示します。デフォルトは False です。

応答オブジェクト：新しく作成されたワークスペースオブジェクトを返します。

サンプル python コード

```
 name = 'test'
 scope_id = '5ce480cc497d4f1b4b9a9e8d'
 filter_id = '5ce480cd497d4f1b4b9a9ea4'
 application = {
     'app_scope_id': scope_id,
     'name': name,
     'absolute_policies': [
         {
      # consumer/provider filter IDs can be ID of a cluster identified during automatic
policy discovery (formerly known as ADM),
       # user inventory filter or app scope.
             'provider_filter_id': filter_id,
             'consumer_filter_id': filter_id,
             'action': 'ALLOW',
       # ALLOW policy for TCP on port 80.
             'l4_params': [
                 {
                     'proto': 6, # TCP
                     'port': [80, 80],  # port range
                 }
             ],
         }
     ],
     'catch_all_action': 'ALLOW'
 }
 restclient.post('/applications', json_body=json.dumps(application))
```

新しいバージョンのインポート

ポリシーをインポートし、ワークスペース（「アプリケーション」）の新しい v* バージョンを作成します。

```
POST /openapi/v1/applications/{application_id}/import
```

パラメータは、上記のワークスペース エンドポイントの作成と同じです。

応答オブジェクト：ワークスペースオブジェクトを返します。

一連のポリシーの検証

新しいバージョンを作成せずに一連のポリシーを検証します。

```
POST /openapi/v1/applications/validate_policies
```

app_scope_id は必須です。残りのパラメータは、前述のワークスペースエンドポイントの作成と同じです。

応答オブジェクト：

属性	タイプ	説明
有効な	boolean	ポリシーが有効かどうかを示します。
errors	アレイ	無効な場合、エラーの詳細

ワークスペースの削除

ワークスペース（「アプリケーション」）を削除します。

```
DELETE /openapi/v1/applications/{application_id}
```

ワークスペースを削除する前に、ワークスペースで適用を無効にする必要があります。

ワークスペースまたはそのクラスタが他のアプリケーションによって（提供されるサービスの関係を介して）使用されている場合、このエンドポイントは `422 Unprocessable Entity` を返します。返されるエラーオブジェクトには、`details` 属性が含まれ、依存オブジェクトの数とともに各タイプの最初の 10 個の ID が示されます。この情報を使用して、問題となっている依存関係を見つけて削除できます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

応答オブジェクト：なし

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
restclient.delete('/applications/%s' % application_id)
```

ワークスペースの更新

このエンドポイントにより、既存のワークスペース（「アプリケーション」）が更新されます。

```
PUT /openapi/v1/applications/{application_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
name	string	（オプション）ワークスペースの更新された名前。
descrip	string	（オプション）ワークスペースの更新された説明。
プライマリ	string	（オプション）ワークスペースをプライマリにするには、「true」に設定します。ワークスペースをセカンダリにするには、「false」に設定します。

応答オブジェクト：指定された ID の更新されたワークスペースオブジェクト。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
req_payload = {
      'name': 'Updated Name',
      'description': 'Updated Description',
      'primary': 'true'
}
resp = restclient.put('/applications/%s' % application_id,
                      json_body=json.dumps(req_payload))
```

ワークスペースの詳細を取得

このエンドポイントは、ワークスペースの完全なエクスポート JSON ファイルを返します。これには、ポリシーとクラスタの定義が含まれます。

```
GET/openapi/v1/applications/{application_id}/details
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
version	string	（オプション）「v10」または「p10」形式のバージョンで、デフォルトは「最新」です。

応答オブジェクト：指定されたワークスペースバージョンのクラスタとポリシーを返します。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
# For v* version v10 and for p* version p10
version = 'v10'
resp = restclient.get('/applications/%s/details?version=%s' % (application_id, version))
```

ワークスペースバージョンの一覧表示

このエンドポイントは、特定のワークスペースのすべてのバージョンのリストを返します。

```
GET /openapi/v1/applications/{application_id}/versions
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
created_before	integer	（オプション）ページネーションの場合、前の応答からの最新バージョンの「created_at」（作成日）に設定します。
limit	integer	（オプション）返される結果の最大数。デフォルトは 50 です。

応答オブジェクト：次の属性を持つオブジェクトの配列：

属性	タイプ	説明
version	string	「v10」または「p10」の形式のバージョン。

属性	タイプ	説明
created_At	integer	ワークスペースが作成されたときのUnixタイムスタンプ。
説明	string	ユーザーが提供する説明。
name	string	表示名。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
created_before = 1612325705
limit = 10
resp = restclient.get('/applications/%s/versions?created_before=%s&limit=%s' %
                      (application_id, created_before, limit))
```

ワークスペースバージョンの削除

このエンドポイントは、クラスタとポリシーを含む指定されたバージョンを削除します。適用または分析されたバージョンは削除されません。メンバーが外部ポリシーを介して別のワークスペースによって参照されている場合、応答は参照のリストと共にエラーを返します。

```
DELETE /openapi/v1/applications/{application_id}/versions/{version}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
version	string	「v10」または「p10」の形式のバージョン。

応答オブジェクト：なし

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
version = 'v10'
resp = restclient.delete('/applications/%s/versions/%s' %
                         (application_id, version))
```

ワークスペースバージョンの比較

このエンドポイントは、提供されたワークスペースバージョン間の相違を計算します。追加されたポリシー、削除されたポリシー、およびオプションで変更されていないポリシーを返します。両方のバージョンに、一致する consistent_uuid によって定義されたクラスタが存在し、クエリが変更された場合、クラスタの変更が含まれます。

```
GET /openapi/v1/applications/{application_id}/version_diff
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
base_version	string	「v10」または「p10」などの完全バージョン。
draft_version	string	「v10」または「p10」などの完全バージョン。
include_unchanged	boolean	デフォルトは false です。変更されていないポリシーを応答で返します。

応答オブジェクト：次の属性を持つオブジェクトが返されます。

属性	タイプ	説明
クラスタ	アレイ	バージョン間で変更されたクラスター。
ポリシー	アレイ	バージョン間で変更されたポリシー。

最新のポリシーの分析

ワークスペース内で一連の最新のポリシー分析を有効にします。

```
POST /openapi/v1/applications/{application_id}/enable_analysis
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

パラメータ：オプションの JSON クエリ本体には、次のキーが含まれます

名前	タイプ	説明
action_note	string	（オプション） ポリシー公開アクションの理由。
name	string	（オプション） 公開されたポリシーバージョンの名前。

名前	タイプ	説明
説明	string	（オプション） 公開されたポリシーバージョンの説明。

応答オブジェクト：次の属性を持つオブジェクトが返されます。

属性	タイプ	説明
data_set	オブジェクト	データセットのJSONオブジェクト表現。
analyzed_policy_version	integer	ワークスペースの分析されたp* バージョン。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
req_payload = {
  'action_note': 'Policy analysis',
  'name': 'Test run 1',
  'description': 'New workloads added.'
}
resp = restclient.post('/applications/%s/enable_analysis' % application_id,
                       json_body=json.dumps(req_payload))
```

単一のワークスペースでポリシー分析を無効にする

ワークスペースでポリシー分析を無効にします。

```
POST /openapi/v1/applications/{application_id}/disable_analysis
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

応答オブジェクト：次の属性を持つオブジェクトを返します。

属性	タイプ	説明
data_set	オブジェクト	データセットのJSONオブジェクト表現。
analyzed_policy_version	integer	最後に分析されたワークスペースの p* バージョン。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
resp = restclient.post('/applications/%s/disable_analysis' % application_id)
```

単一のワークスペースの適用

ワークスペースで最新のポリシーセットの適用を有効にします。

```
POST /openapi/v1/applications/{application_id}/enable_enforce
```



警告 新しいホスト ファイアウォール ルールが挿入され、関連するホスト上で既存のルールがすべて削除されます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
version	string	(オプション)適用するポリシーのバージョン。

バージョンが指定されていない場合、ワークスペースの最新のポリシーが適用されます。バージョンは「p*」の形式で指定することが推奨されます。整数のみが指定されている場合、対応する「p*」バージョンが適用されます。

応答オブジェクト：次の属性を持つオブジェクトを返します。

名前	タイプ	説明
新時代	string	最新の適用プロファイルの固有識別子。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
req_payload = {
  'version': 'p10'
}
resp = restclient.post('/applications/%s/enable_enforce' % application_id,
                       json_body=json.dumps(req_payload))
```

単一ワークスペースの適用の無効化

ワークスペースへの適用を無効にします。

```
POST /openapi/v1/applications/{application_id}/disable_enforce
```

警告 新しいホスト ファイアウォール ルールが挿入され、関連するホスト上で既存のルールがすべて削除されます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

応答オブジェクト：次の属性を持つオブジェクトを返します。

名前	タイプ	説明
新時代	string	最新の適用プロファイルの固有識別子。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
resp = restclient.post('/applications/%s/disable_enforce' %
                       application_id)
```

自動ポリシー検出の開始

ワークスペースのポリシーを自動的に検出します（以前は「ADM 実行の送信」と呼ばれていました）。

```
POST /openapi/v1/applications/{application_id}/submit_run
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
start_time	string	自動ポリシー検出実行の入力時間間隔の開始時刻。
end_time	string	自動ポリシー検出実行の入力時間間隔の終了時刻。

名前	タイプ	説明
clustering_granularity	string	（オプション）クラスタの細分度により、ユーザーは自動ポリシー検出によって生成されるクラスタのサイズを制御できます。想定される値：VERY_FINE、FINE、MEDIUM、COARSE、またはVERY_COARSE
port_generalization	string	（オプション）ポートの一般化により、ポートの汎用化を実行するときに必要な統計的有意性のレベルが制御されます。想定される値：DISABLED、CONSERVATIVE、MODERATE、AGGRESSIVE、またはVERY_AGGRESSIVE
policy_compression	string	（オプション）ポリシー圧縮が有効になっている場合、ワークスペース内に生成されたクラスタの中で、相当な頻度で使用されるポリシー（同じプロバイダーポートを使用するポリシー）は、親に「除外」される可能性があります。つまり、親範囲全体に置き換えられる可能性があります。想定される値：DISABLED、CONSERVATIVE、MODERATE、AGGRESSIVE、またはVERY_AGGRESSIVE
auto_accept_policy_connectors	boolean	（オプション）自動承諾ポリシーコネクタ 自動ポリシー検出中に作成されたすべての発信ポリシー要求は、自動的に受け入れられます。

名前	タイプ	説明
enable_exclusion_filter	boolean	（オプション）除外フィルタの有効化オプションは、ユーザー定義の除外フィルター（存在する場合）のいずれかに一致するすべてのカンバセーションを無視する柔軟な対応を可能にします。詳細については、「除外フィルタ」を参照してください。
enable_default_exclusion_filter	boolean	（オプション）デフォルトの除外フィルタの有効化オプションは、デフォルトの除外フィルタ（存在する場合）のいずれかに一致するすべてのカンバセーションを無視する柔軟な対応を可能にします。詳細については、「デフォルトの除外フィルタ」を参照してください。
enable_service_discovery	boolean	（オプション）エージェントのサービス検出の有効化が設定されている場合、エージェントノードに存在するサービスに関連した一時的なポート範囲情報がレポートされます。次に、レポートされたポート範囲情報に基づいてポリシーが生成されます。
carry_over_policies	boolean	（オプション）承認されたポリシーの引き継ぎが設定されている場合、UI または OpenAPI を使用してユーザーによって承認済みとしてマークされているすべてのポリシーが保持されます。

名前	タイプ	説明
skip_clustering	boolean	（オプション）クラスタリングをスキップしてポリシーを作成が設定されている場合、新しいクラスタは生成されず、既存の承認済みクラスタまたはインベントリフィルタからポリシーが生成されます。設定されていない場合は、範囲内のすべてのワークロードがポリシーに含められます。
deep_policy_generation	boolean	（オプション）単一範囲ではなく、範囲ツリーのブランチのポリシーを生成することができます。詳細については、1つの範囲または範囲ツリーのブランチのポリシーの検出（538 ページ）およびサブトピックを参照してください。
use_default_config	boolean	（オプション）このオプションが設定されている場合、自動ポリシー検出は、以前の実行設定の代わりにデフォルトのポリシー検出設定を使用します。詳細については、「デフォルトのポリシー検出設定」を参照してください。

（注） ワークスペースで自動ポリシー検出が以前に実行された場合、指定されていないオプションパラメータのデフォルト値は、以前の自動ポリシー検出実行設定から取得されます。それ以外の場合、デフォルト値はデフォルトのポリシー検出設定から取得されます。

応答オブジェクト：次の属性を持つオブジェクトを返します。

名前	タイプ	説明
message	string	自動ポリシー検出実行の成功/失敗に関するメッセージ。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
req_payload = {
      'start_time': '2020-09-17T10:00:00-0700',
      'end_time': '2020-09-17T11:00:00-0700',
      # Optional Parameters.
      'clustering_granularity': 'FINE',
      'port_generalization': 'AGGRESSIVE',
      'policy_compression': 'AGGRESSIVE',
      'auto_accept_policy_connectors': False,
      'enable_exclusion_filter': True,
      'enable_default_exclusion_filter': True,
      'enable_service_discovery': True,
      'carry_over_policies': True,
      'skip_clustering': False,
      'deep_policy_generation': True,
      'use_default_config': False
}
resp = restclient.post('/applications/%s/submit_run' % application_id,
                       json_body=json.dumps(req_payload))
```

ポリシー検出実行のステータスの取得

ワークスペースで実行されている自動ポリシー検出のステータスをクエリします。

```
GET /openapi/v1/applications/{application_id}/adm_run_status
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

応答オブジェクト：次の属性を持つオブジェクトを返します。

名前	タイプ	説明
status	string	自動ポリシー検出実行のステータス。値：PENDING、COMPLETE、または FAILED

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
resp = restclient.get('/applications/%s/adm_run_status' % application_id)
```

ポリシー

この一連の API を使用して、ポリシーの追加、編集、または削除を管理できます。 create および update catch all アクションには `version` パラメータが必要です。API キーに関連付けられた `user_role_scope_management` 機能が必要です。

ポリシー オブジェクト

ポリシーオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	ポリシーの一意の識別子。
application_id	string	ポリシーが属するワークスペースの ID。
consumer_filter_id	string	定義されたフィルタの ID。現在、任意のクラスタ、ユーザー定義フィルタ、または範囲をポリシーのコンシューマとして使用できます。
provider_filter_id	string	定義されたフィルタの ID。現在、任意のクラスタ、ユーザー定義フィルタ、または範囲をポリシーのプロバイダーとして使用できます。
version	string	ポリシーが属するワークスペースのバージョンを示します。
rank	string	ポリシーランク。入力可能値：DEFAULT、ABSOLUTE、CATCHALL。
policy_action	string	入力可能値は、ALLOW または DENY です。コンシューマとプロバイダー間の特定のサービスポートまたはプロトコルについて、トラフィックを許可するかドロップするかを示します。
priority	integer	ポリシーを並べ替えるために使用されます。
l4_params	l4params の配列	許可されたポートとプロトコルのリスト。

L4Params オブジェクト属性：

名前	タイプ	説明
proto	integer	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
port	array	ポートの包含範囲。例：[80, 80] または [5000, 6000]。
説明	string	このプロトとポートに関する短い文字列。
approved	boolean	ポリシーがユーザーによって承認されている場合。

ポリシーの取得

このエンドポイントは、特定のワークスペースのポリシーのリストを返します。このAPIは、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/policies
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
version	string	ポリシーを取得するワークスペースのバージョンを示します。
consumer_filter_id	string	（オプション）コンシューマフィルタ ID で出力をフィルタ処理します。
provider_filter_id	string	（オプション）コンシューマフィルタ ID で出力をフィルタ処理します。

ポリシー ID は、バージョンごとに変更できます。公開バージョンからポリシーのリストを取得するには、バージョン番号の前に「p」を付ける必要があります。たとえば、公開されたバージョン 3 のすべてのポリシーを取得するには、次のような要求を実行できます。

```
GET /openapi/v1/applications/{application_id}/policies?version=p3
```

以下に示すように、この特定のワークスペース内のすべてのポリシーのオブジェクトを返します。

```
{
  absolute_policies: [ ... ],
  default_policies: [ ... ],
```

```
      catch_all_action:
    }
```

サンプル python コード

```
application_id = '5f88c996755f023f3bafe163'
restclient.get('/applications/%s/policies' % application_id, params={'version': '1'})
```

デフォルトポリシーの取得

このエンドポイントは、特定のワークスペースのデフォルトポリシーのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/default_policies
```

パラメータ：

名前	タイプ	説明
id	string	ポリシーの一意の識別子。
version	string	ポリシーを取得するワークスペースのバージョンを示します。
limit	integer	要求あたりのポリシー数を制限します。
offset	integer	（オプション）以前の応答から受け取ったオフセット番号。常に `limit` と共に使用する必要があります。
consumer_filter_id	string	（オプション）コンシューマフィルタ ID で出力をフィルタ処理します。
provider_filter_id	string	（オプション）プロバイダーフィルタ ID で出力をフィルタ処理します。

このワークスペースの提供されたバージョンのデフォルトポリシーのリストを返します。応答には、要求された数のポリシーと `offset` が含まれており、次のポリシーのセットを取得するには、後続の要求でこの `offset` を使用します。応答に `offset` がない場合は、すべてのポリシーがすでに取得されていることを示します。

サンプル python コード

```
application_id = '5f88c996755f023f3bafe163'
restclient.get('/applications/%s/default_policies' % application_id, params={'version':
'1', 'limit': 3, 'offset': 3})
```

サンプル応答

```
{
  "results": [
  PolicyObject4,
  PolicyObject5,
  PolicyObject6
  ],
  "offset": 6
}
```

絶対ポリシーの取得

このエンドポイントは、特定のワークスペース内の絶対ポリシーのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/absolute_policies
```

パラメータ：

名前	タイプ	説明
version	string	ポリシーを取得するワークスペースのバージョンを示します。
limit	integer	要求あたりのポリシー数を制限します。
offset	integer	（オプション）以前の応答から受け取ったオフセット番号。常に `limit` と共に使用する必要があります。
consumer_filter_id	string	（オプション）コンシューマフィルタ ID で出力をフィルタ処理します。
provider_filter_id	string	（オプション）プロバイダーフィルタ ID で出力をフィルタ処理します。

このワークスペースの提供されたバージョン内の絶対ポリシーのリストを返します。応答には、要求された数のポリシーと `offset` が含まれており、次のポリシーのセットを取得するには、後続の要求でこの `offset` を使用します。応答に `offset` がない場合は、すべてのポリシーがすでに取得されていることを示します。

サンプル python コード

```
 application_id = '5f88c996755f023f3bafe163'
 restclient.get('/applications/%s/absolute_policies' % application_id, params={'version':
'1', 'limit': 3})
```

サンプル応答

```
{
  "results": [
  PolicyObject1,
  PolicyObject2,
  PolicyObject3
  ],
  "offset": 3
}
```

Catch All ポリシーの取得

このエンドポイントは、特定のワークスペースの Catch All ポリシーを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/catch_all
```

パラメータ：

名前	タイプ	説明
version	string	ポリシーを取得するワークスペースのバージョンを示します。

ワークスペースの指定されたバージョンの Catch All ポリシーを表す単一のポリシーオブジェクトを返します。

サンプル python コード

```
application_id = '5f88c996755f023f3bafe163'
restclient.get('/applications/%s/catch_all' % application_id, params={'version': '1'})
```

特定のポリシーの取得

このエンドポイントは、ポリシーのインスタンスを返します。

```
GET /openapi/v1/policies/{policy_id}
```

指定した ID に関連付けられているポリシーオブジェクトを返します。

サンプル python コード

```
policy_id = '5f88ca1e755f0222f85ce85c'
restclient.get('/policies/%s' % policy_id)
```

ポリシー識別子を使用した特定のポリシーの検索

このエンドポイントは、ポリシー識別子パラメータを複合キーとして使用して、特定のポリシーを検索します。

```
POST /openapi/v1/policies/search
```

クエリ本文は、次のスキーマを使用した JSON 本文で構成されます。

名前	タイプ	説明
application_id	string	アプリケーション ワークスペースの ID。
policy_identifier	オブジェクト	一貫したポリシー識別子を構成するフィールド。

ポリシー識別子フィールドは、次のスキーマを使用して構成されています。

名前	タイプ	説明
version	string	（オプション）ポリシーを取得するアプリケーションのバージョンを示します。指定しない場合、デフォルトでアプリケーションの最新の「v」バージョンに設定されます。
consumer_consistent_uuid	string	コンシューマまたは送信元の一貫した UUID。
provider_consistent_uuid	string	プロバイダーまたは宛先の一貫した UUID。
rank	string	ポリシーランクは「DEFAULT」または「ABSOLUTE」のいずれかである必要があります。
action	string	ポリシーアクションは、「ALLOW」または「DENY」のいずれかである必要があります。
priority	integer	ポリシーの優先順位を示す値。
protocol	integer	IP プロトコル番号（0 ～ 255）。
start_port	integer	（オプション）ポート範囲の開始（0 ～ 65535）。指定されていない場合、デフォルトは 0 です。
end_port	integer	（オプション）ポート範囲の終了（0 ～ 65535）。start_port が 0 の場合、デフォルトは 65535 です。そうでない場合は start_port です。

サンプル python コード

```
application_id = '5f88ca1e755f0222f85ce85c'
consumer_id = '5f88ca1e755f0222f85ce85d'
provider_id = '5f88ca1e755f0222f85ce85d'
rank = 'DEFAULT'
action = 'ALLOW'
priority = 100
protocol = 6
start_port = 80
version = 'p3'

req_body = f'''
{{
    "application_id": "{application_id}",
    "policy_identifier": {{
        "consumer_consistent_uuid": "{consumer_id}",
        "provider_consistent_uuid": "{provider_id}",
        "rank": "{rank}",
        "action": "{action}",
        "priority": {priority},
        "protocol": "{protocol}",
        "start_port": "{start_port}",
        "version": "{version}"
    }}
}}'''
restclient.post('/policies/search', json_body=req_body)
```

ポリシーの作成

このエンドポイントは、新しいポリシーを作成するために使用されます。

```
POST /openapi/v1/applications/{application_id}/policies
```

パラメータ：

属性	タイプ	説明
consumer_filter_id	string	定義されたフィルタの ID。
provider_filter_id	string	定義されたフィルタの ID。
version	string	ポリシーを更新するワークスペースのバージョンを示します。
rank	string	値は、ランク付けの DEFAULT、ABSOLUTE、または CATCHALL になります。
policy_action	string	値は ALLOW または DENY です。つまり、指定されたサービスポートまたはプロトコルで、コンシューマからプロバイダーへのトラフィックを許可またはドロップします。

属性	タイプ	説明
priority	integer	ポリシーを並べ替えるために使用されます。

サンプル python コード

```
  req_payload = {
    "version": "v1",
    "rank" : "DEFAULT",
    "policy_action" : "ALLOW",
    "priority" : 100,
    "consumer_filter_id" : "123456789",
    "provider_filter_id" : "987654321",
  }
  resp = restclient.post('/openapi/v1/applications/{application_id}/policies',
json_body=json.dumps(req_payload))
```

デフォルトポリシーの作成

このエンドポイントは、新しいデフォルトポリシーを作成するために使用されます。このエンドポイントでは、ポリシーエンドポイントの作成と同様のデフォルトポリシーが作成されます。

```
POST /openapi/v1/applications/{application_id}/default_policies
```

絶対ポリシーの作成

このエンドポイントは、新しい絶対ポリシーを作成するために使用されます。このエンドポイントでは、ポリシーエンドポイントの作成と同様の絶対ポリシーが作成されます。

```
POST /openapi/v1/applications/{application_id}/absolute_policies
```

ポリシーの更新

このエンドポイントはポリシーを更新します。

```
PUT /openapi/v1/policies/{policy_id}
```

パラメータ：

属性	タイプ	説明
consumer_filter_id	string	定義されたフィルタの ID。
provider_filter_id	string	定義されたフィルタの ID。

属性	タイプ	説明
policy_action	string	入力可能値は、ALLOW または DENY です。コンシューマとプロバイダー間の特定のサービスポートまたはプロトコルについて、トラフィックを許可するかドロップするかを示します。
priority	integer	ポリシーの優先順位の並べ替えに使用

指定された ID に関連付けられている変更されたポリシーオブジェクトを返します。

Catch All の更新

このエンドポイントは、特定のワークスペースの Catch All を更新します。

```
PUT /openapi/v1/applications/{application_id}/catch_all
```

パラメータ：

属性	タイプ	説明
version	string	ポリシーを更新するワークスペースのバージョンを示します。
policy_action	string	入力可能値は、ALLOW または DENY です。このワークスペースのいずれのポリシーにも一致しないトラフィックを許可するか、ドロップするかを示します。

ポリシーへのサービスポートの追加

このエンドポイントは、特定のポリシーのサービスポートを作成するために使用されます。

```
POST /openapi/v1/policies/{policy_id}/l4_params
```

パラメータ：

属性	タイプ	説明
version	string	ポリシーを取得するワークスペースのバージョンを示します。

属性	タイプ	説明
start_port	integer	範囲の開始ポート。
end_port	integer	範囲の終了ポート。
proto	integer	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
説明	string	（オプション）このプロトコルとポートに関する短い文字列。

ポリシーのサービスポートの更新

このエンドポイントは、ポリシーの指定されたサービスポートを更新します。

```
PUT /openapi/v1/policies/{policy_id}/l4_params/{l4_params_id}
```

パラメータ：

属性	タイプ	説明
approved	bool	ポリシーを承認済みとしてマークします。

ポリシーのサービスポートの削除

このエンドポイントは、ポリシーの指定されたサービスポートを削除します。（オプション）詳細については、「除外フィルタ」を参照してください。

```
DELETE /openapi/v1/policies/{policy_id}/l4_params/{l4_params_id}
```

パラメータ：

属性	タイプ	説明
create_exclusion_filter	bool	（オプション）trueの場合、ポリシーに一致する除外フィルタが作成されます。このフィルタに一致するフローは、今後の自動ポリシー検出の実行から除外されます。詳細については、「除外フィルタ」を参照してください。

ポリシーの削除

このエンドポイントは、指定されたポリシーを削除します。除外フィルタは作成されません。

```
DELETE /openapi/v1/policies/{policy_id}
```

識別子を持つポリシーの削除

このエンドポイントは、ポリシー識別子パラメータを使用して、指定されたポリシーを削除します。除外フィルタは作成されません。

```
DELETE /openapi/v1/policies/destroy_with_identifier
```

クエリ本文は、次のスキーマを使用した JSON 本文で構成されます。

名前	タイプ	説明
application_id	string	アプリケーション ワークスペースの ID。
policy_identifier	オブジェクト	一貫したポリシー識別子を構成するフィールド。

ポリシー識別子フィールドは、次のスキーマを使用して構成されています。

名前	タイプ	説明
version	string	（オプション）削除操作を実行するアプリケーション ワークスペースの「v」バージョン。指定されていない場合、デフォルトはワークスペースの最新の「v」バージョンです。
consumer_consistent_uuid	string	コンシューマまたは送信元の一貫した UUID
provider_consistent_uuid	string	プロバイダーまたは宛先の一貫した UUID
rank	string	ポリシーランクは「DEFAULT」または「ABSOLUTE」のいずれかである必要があります。
action	string	ポリシーアクションは、「ALLOW」または「DENY」のいずれかである必要があります。

名前	タイプ	説明
priority	integer	ポリシーのプライオリティ値。
protocol	integer	ポリシーの IP プロトコル番号（0 ～ 255）
start_port	integer	（オプション）ポート範囲の開始（0 ～ 65535）。指定されていない場合、デフォルトは 0 です
end_port	integer	（オプション） ポート範囲の終了（0 ～65535）。start_port が 0 の場合、デフォルトは 65535 で、そうでない場合は start_port です。

サンプル python コード

```
application_id = '5f88ca1e755f0222f85ce85c'
consumer_id = '5f88ca1e755f0222f85ce85d'
provider_id = '5f88ca1e755f0222f85ce85d'
action = 'ALLOW'
rank = 'DEFAULT'
protocol = 6
start_port = 80
priority = 100
version = '5'

req_body = f'''
{{
    "application_id": "{application_id}",
    "policy_identifier": {{
        "consumer_consistent_uuid": "{consumer_id}",
        "provider_consistent_uuid": "{provider_id}",
        "rank": "{rank}",
        "priority": {priority},
        "action": "{action}",
        "protocol": "{protocol}",
        "start_port": "{start_port}",
        "version": "{version}"
    }}
}}'''
restclient.delete('/policies/destroy_with_identifier', json_body=req_body)
```

ポリシーの簡易分析

このエンドポイントを使用して、ルート範囲で分析または適用されたポリシーと一致する仮想フローの一連のポリシーを見つけることができます。詳細については、「簡易分析」を参照してください。

この API は、ルート範囲への最小限の読み取りアクセス権を持つユーザーのみが使用できます。また、API キーに関連付けられた `app_policy_management` 機能が必要です。

```
POST /openapi/v1/policies/{rootScopeID}/quick_analysis
```

クエリ本文は、次のスキーマを使用した JSON 本文で構成されます。

名前	タイプ	説明
consumer_ip	string	クライアント/コンシューマの IP アドレス。
provider_ip	string	サーバー/プロバイダーの IP アドレス。
provider_port	integer	（オプション）プロバイダーポート。TCP または UDP フローにのみ関連します。
protocol	string	フローのプロトコル（TCP など）。
analysis_type	string	分析タイプは、**analyzed** または **enforced** にできます。分析タイプが「analyzed」の場合、ルート範囲で分析されたすべてのポリシーに対してフローを照合することによって、フローが決定されます。分析タイプが「enforced」の場合、ルート範囲で適用されたすべてのポリシーに対してフローを照合することによって、フローが決定されます。

名前	タイプ	説明
application_id	string	（オプション）プライマリワークスペースの ID は常にワークスペース「v」バージョンを伴います。指定されている場合は、ルート範囲内の他のワークスペースからの分析および適用済みポリシーと共に、指定されたバージョンのポリシーを使用してフローが決定されます。このフィールドが省略された場合、ルート範囲内のすべての分析および適用済みポリシーを考慮して、フローが決定されます。
version	整数	（オプション）上記のワークスペースの「v」バージョン。これは、application_id が指定されている場合は指定する必要があります。それ以外の場合は省略する必要があります。

サンプル要求

要求の本文は、JSON 形式のクエリである必要があります。

分析されたすべてのポリシーに基づいてフローが決定されるクエリ本文の例は、次のようになります。

```
    req_payload = {
        "consumer_ip": "4.4.1.1",
        "provider_ip": "4.4.2.1",
        "provider_port": 9081,
        "protocol": "TCP",
        "analysis_type": "analyzed"
    }
    resp = restclient.post('/openapi/v1/policies/{rootScopeID}/quick_analysis',
json_body=json.dumps(req_payload))
```

フローの決定が、ワークスペースの 'v' バージョンのポリシーと、ルート範囲内の他のすべてのワークスペースから分析されたポリシーに基づくクエリ本文の例は、次のようになります。

```
    req_payload = {
        "consumer_ip": "4.4.1.1",
        "provider_ip": "4.4.2.1",
        "provider_port": 9081,
        "protocol": "TCP",
        "analysis_type": "analyzed",
        "application_id": "5e7e5f56497d4f0bc26c7bb3",
```

```
        "version": 1
    }
    resp = restclient.post('/openapi/v1/policies/{rootScopeID}/quick_analysis',
json_body=json.dumps(req_payload))
```

サンプル応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

キー	値
policy_decision	許可または拒否されるかどうかの仮想フローの決定。
outbound_policy	発信トラフィックを許可/拒否するコンシューマのポリシー
inbound_policy	着信トラフィックを許可/拒否するプロバイダーのポリシー

```
{
    "policy_decision": "ALLOW",
    "outbound_policy": {
        "policy_rank": "DEFAULT",
        "start_port": 9082,
        "l4_detail_id": "5e7e600f497d4f7341f4f6d0",
        "src_filter_id": "5e7e600e497d4f7341f4f459",
        "end_port": 9082,
        "cluster_edge_id": "5e7e600f497d4f7341f4f6d1",
        "dst_filter_id": "5e7d0efc497d4f44b6b09351",
        "action": "ALLOW",
        "protocol": "TCP",
        "app_scope_id": "5e7e5f3a497d4f0bc26c7bb0"
    },
    "inbound_policy": {
        "policy_rank": "DEFAULT",
        "start_port": 9082,
        "l4_detail_id": "5e7e600f497d4f7341f4f6d0",
        "src_filter_id": "5e7e600e497d4f7341f4f459",
        "end_port": 9082,
        "cluster_edge_id": "5e7e600f497d4f7341f4f6d1",
        "dst_filter_id": "5e7d0efc497d4f44b6b09351",
        "action": "ALLOW",
        "protocol": "TCP",
        "app_scope_id": "5e7e5f3a497d4f0bc26c7bb0"
    }
}
```

ポリシー統計情報

このエンドポイントは、ある時間間隔でポリシーについて観察されたパケット、バイト、およびカンバセーションの数を返します。カンバセーションは、概して言えば1時間の実行間隔で集約されたポリシーに一致するフロー監視になります。1 時間以内に特定のポリシーについて測定されたカンバセーションの数は、その1時間にネットワークを介して通信したコンシューマとプロバイダーのインベントリ項目の個別のペアの数を表します。

このエンドポイントはポリシー識別子パラメータを入力として受け入れますが、ワークスペースの公開されたバージョンからのポリシーおよびL4パラメータIDの使用が強く推奨されます。

（注） アプリケーションワークスペースの新しいバージョンが公開された後、その結果が利用可能になるまでに最大6時間かかることがあります。すべてのタイムスタンプの解決にも、6時間の最小実行間隔があります。

アプリケーションワークスペースの適用されたバージョン全体におけるポリシーの、ポリシー統計を取得するURLパスは次のとおりです。

```
POST /openapi/v1/policies/stats/enforced
```

アプリケーションワークスペースの分析されたバージョン全体におけるポリシーの、ポリシー統計を取得するURLパスは次のとおりです。

```
POST /openapi/v1/policies/stats/analyzed
```

クエリ本文は、次のスキーマを使用したJSON本文で構成されます。

表 *55：*

名前	タイプ	説明
application_id	string	アプリケーション ワークスペースのID。
t0	string	RFC-3339形式の時間間隔の開始。
t1	string	（オプション）RFC-3339形式の時間間隔の終了。指定しない場合、デフォルトは現在の時間です。
policy_id	string	ポリシーのID。ポリシー識別子が存在する場合は不要です。
l4_param_id	string	L4パラメータのID。ポリシー識別子が存在する場合、または「CATCH_ALL」ポリシーの場合は不要です。
policy_identifier	オブジェクト	一貫したポリシー識別子を構成するフィールド。

ポリシー識別子フィールドは、次のスキーマを使用して構成されています。

名前	タイプ	説明
consumer_consistent_uuid	string	コンシューマまたは送信元の一貫した UUID。
provider_consistent_uuid	string	プロバイダーまたは宛先の一貫した UUID。
rank	string	ポリシーランクは「DEFAULT」または「ABSOLUTE」のいずれかである必要があります。
action	string	ポリシーアクションは、「ALLOW」または「DENY」のいずれかである必要があります。
priority	integer	ポリシーの優先順位を示す値。
protocol	integer	ポリシーの IP プロトコル番号（0 ～ 255）。
start_port	integer	（オプション）ポート範囲の開始（0 ～ 65535）。指定されていない場合、デフォルトは 0 です
end_port	integer	（オプション）ポート範囲の終了（0 ～65535）。start_port が 0 の場合、デフォルトは 65535 です。そうでない場合は start_port です。

サンプル python コード

```
application_id = '5f88ca1e755f0222f85ce85c'
consumer_id = '5f88ca1e755f0222f85ce85d'
provider_id = '5f88ca1e755f0222f85ce85d'
action = 'ALLOW'
rank = 'DEFAULT'
protocol = 6
start_port = 80
priority = 100

req_body = f'''
{{
    "application_id": "{application_id}",
    "t0":"2022-07-06T00:00:00Z",
    "t1":"2022-07-28T19:00:00Z",
    "policy_identifier": {{
```

```
        "consumer_consistent_uuid": "{consumer_id}",
        "provider_consistent_uuid": "{provider_id}",
        "rank": "{rank}",
        "priority": {priority},
        "action": "{action}",
        "protocol": "{protocol}",
        "start_port": "{start_port}"
    }}
}}'''
restclient.post('/policies/stats/analyzed', json_body=req_body)

# For CATCH_ALL policies:
root_app_scope_id = '6f88ca1e755f0222f85ce85e'
rank = 'CATCH_ALL'
action = 'DENY'
req_body = f'''
{{
    "application_id": "{application_id}",
    "t0":"2022-07-06T00:00:00Z",
    "t1":"2022-07-28T19:00:00Z",
    "policy_identifier": {{
        "consumer_consistent_uuid": "{root_app_scope_id}",
        "provider_consistent_uuid": "{root_app_scope_id}",
        "rank": "{rank}",
        "action": "{action}"
    }}
}}'''

restclient.post('/policies/stats/analyzed', json_body=req_body)
```

サンプル応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

表 *56 :*

キー	値
conversation_count	指定された期間とポリシーで観察されたカンバセーションの数。
packet_count	指定された期間とポリシーで観察されたパケットの数。
byte-count	指定された期間とポリシーで観察されたバイト数。
first_seen_at	このポリシーのフローを最初に観察したときのタイムスタンプ（RFC-3339 形式）。
last_seen_at	このポリシーのフローを最後に観察したときのタイムスタンプ（RFC-3339 形式）。
agg_start_version	時間 t0 以降に記録された、このポリシーの最も古い公開バージョン。
agg_start_time	agg_start_version が公開されたタイムスタンプ。

```
{
  "conversation_count": 72,
  "packet_count": 800,
  "byte_count": 1960,
  "first_seen_at": "2022-09-09T11:00:00.000Z",
  "last_seen_at": "2022-09-09T11:00:00.000Z",
  "agg_start_version": 4,
  "agg_start_time": "2022-08-10T23:00:00.000Z"
}
```

未使用のポリシー

このエンドポイントは、指定された時間間隔でカンバセーションが観察されなかった公開ワークスペース内のポリシー識別子を返します。

ポリシー識別子

すべてのポリシーと ADM で生成されたクラスタは、基になるフィルタクエリまたはポリシーのポートとプロトコルが変更されなくても、複数のアプリケーション ワークスペース バージョン間でその ID を変更できます。ワークスペースバージョン間で特定のポリシーのフローヒット数を追跡するため、バージョン間で変更されないフィルタには一貫したUUIDを使用し、ポリシー識別子と呼ばれる複合キーを使用します。これには、プロバイダーとコンシューマの一貫した UUID に加え、ランク、アクション、優先順位、ポート、プロトコルが含まれます。

したがって、ポリシー識別子は、アプリケーション ワークスペースのバージョン間でポリシーの重要な側面を識別および説明できる複合キーとして機能しますが、ポリシー ID（通常の CRUDエンドポイントで使用されるものなど）はワークスペースのバージョン間で変更できます。

（注） プロバイダー/コンシューマの一貫性のあるUUIDとポリシー識別子は、さまざまなアプリケーション ワークスペース バージョン間で共有されるため、フィルタまたはポリシーを一意に識別しません。

特定のクラスタまたはポリシーで CRUD 操作を実行するには、検索エンドポイントで識別子を特定のアプリケーション ワークスペース バージョンの具体的なポリシーに解決することを推奨します。

通常の CRUD 操作はポリシー ID を使用して実行できますが、ポリシー統計情報と識別子による破壊のみで、ポリシー識別子を入力として使用できます。これは主に、便宜上検索のための中間呼び出しを避けるためであり、代わりにワークスペース内の未使用のポリシーを直接検証して破壊します。

ポリシー ID とフィルタ ID を可能な限り使用し、ポリシー統計情報または識別子による破壊 APIエンドポイント用に手動でポリシー識別子を生成しないことを強く推奨します。ただし、次の例は、ポリシーオブジェクトからポリシー識別子を生成する1つの方法を示しています。

```
resp = restclient.get(f'/policies/631b0590497d4f09b537b973')
policy = resp.json() # policy object
policy_identifer = {
    'consumer_consistent_uuid': policy['consumer_filter']['consistent_uuid'],
```

```
        'provider_consistent_uuid': policy['provider_filter']['consistent_uuid'],
        'rank': policy['rank'],
        'action': policy['action'],
        'priority': policy['priority'],
        'protocol': policy['l4_params'][0]['proto'],
        'start_port': policy['l4_params'][0]['port'][0],
        'end_port': policy['l4_params'][0]['port'][1]
    }
```



（注） アプリケーションワークスペースの新しいバージョンが公開された後、その結果が利用可能になるまでに最大6時間かかることがあります。各タイムスタンプの解決にも、最小6時間を必要とします。

アプリケーションワークスペースの適用されたバージョン間で未使用のポリシーを取得するためのURLパスは次のとおりです。

```
POST /openapi/v1/unused_policies/{application_id}/enforced
```

アプリケーションワークスペースの分析されたバージョン間で未使用のポリシーを取得するためのURLパスは次のとおりです。

```
POST /openapi/v1/unused_policies/{application_id}/analyzed
```

クエリ本文は、次のスキーマを使用したJSON本文で構成されます。

名前	タイプ	説明
t0	string	RFC-3339形式の時間間隔の開始。
t1	string	（オプション）RFC-3339形式の時間間隔の終了。指定しない場合、デフォルトは現在の時間です。
limit	integer	（オプション）要求あたりのポリシー数を制限します。
offset	string	（オプション）前の応答から受信したオフセット（ページネーションに有用）。

```
application_id = '62e1915e755f026f2bcdd805'
resp = restclient.post(f'/unused_policies/{application_id}/analyzed', json_body=f'''

{{
    "t0":"2022-07-06T00:00:00Z",
    "t1":"2022-07-28T19:00:00Z"
}}''')'
```

サンプル応答

この応答は、本文のJSONオブジェクトで、次のプロパティがあります。

キー	値
application_id	アプリケーション ワークスペースの ID。
policy_identifiers	未使用のポリシーのポリシー識別子のリスト。
offset	結果の次のページに渡される応答オフセット。

結果の次のページを生成するには、offset の応答で受信したオブジェクトを取得し、それを次のクエリの offset の値として渡します。


```
{
  "application_id": "63054a97497d4f2dc113a9c4",
  "policy_identifiers": [
    {
      "consumer_consistent_uuid": "62fff45c497d4f5064973c4d",
      "provider_consistent_uuid": "62fff45c497d4f5064973c4d",
      "version": "p1",
      "rank": "DEFAULT",
      "policy_action": "ALLOW",
      "priority": 10,
      "proto": 6,
      "start_port": 10000,
      "end_port": 10000,
      "agg_start_version": 1,
      "agg_start_time": "2022-08-10T23:00:00.000Z"
    },
    {
      "consumer_consistent_uuid": "62fff45c497d4f5064973c4d",
      "provider_consistent_uuid": "62fff45c497d4f5064973c4d",
      "version": "p1",
      "rank": "DEFAULT",
      "policy_action": "ALLOW",
      "priority": 10,
      "protocol": 6,
      "start_port": 10001,
      "end_port": 10001,
      "agg_start_version": 1,
      "agg_start_time": "2022-08-10T23:00:00.000Z"
    }
  ],
  "offset": "eyJvZmZzZXQiOjZ9"
}
```


ポリシーテンプレート

この一連の API は、ポリシーテンプレートを追加、編集、または削除するために使用できます。API キーに関連付けられている app_policy_management 機能が必要です。

ポリシーテンプレートの取得

このエンドポイントは、特定のルート範囲のポリシーテンプレートのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/application_templates?root_app_scope_id={root_app_scope_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_app_scope_id	string	ルート範囲の一意の識別子。

応答オブジェクト：指定されたルート範囲のポリシー テンプレート オブジェクトのリストを返します。

サンプル python コード

```
root_app_scope_id = '<root-app-scope-id>'
restclient.get('/application_templates?root_app_scope_id=%s' % root_app_scope_id)
```

特定のポリシーテンプレートの取得

このエンドポイントは、ポリシーテンプレートのインスタンスを返します。

```
GET /openapi/v1/application_templates/{template_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
template_id	string	ポリシーテンプレートの固有識別子。

応答オブジェクト：指定された ID のポリシー テンプレート オブジェクトを返します。

サンプル python コード

```
template_id = '<template-id>'
restclient.get('/application_templates/%s' % template_id)
```

ポリシーテンプレートの作成

このエンドポイントは、新しいポリシーテンプレートを作成するために使用されます。

```
POST /openapi/v1/application_templates
```

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
name	string	インポート時にテンプレートの名前として使用されます。
説明	string	（オプション）適用プロセス中に表示されるテンプレートの説明

属性	タイプ	説明
パラメータ	パラメータオブジェクト	テンプレートパラメータ（下記を参照）。
absolute_policies	ポリシーオブジェクトの配列	（オプション）絶対ポリシーの配列。
default_policies	ポリシーオブジェクトの配列	（必須）デフォルトポリシーの配列、空にすることが可能。

応答オブジェクト: 作成されたポリシーテンプレート オブジェクトを返します。

サンプル python コード

```
root_app_scope_id = '<root-app-scope-id>'
payload = {'root_app_scope_id': root_app_scope_id,
           'name': "policy_name",
           'default_policies': [
              {
                'action': 'ALLOW',
                'priority': 100,
                'l4_params': [
                  {
                    'proto': 17,
                    'port': [80, 90]
                  }
                ]
              }
            ]
          }
restclient.post('/application_templates',
                json_body=json.dumps(payload))
```

ポリシーテンプレートの更新

このエンドポイントは、ポリシーテンプレートを更新します。

```
PUT /openapi/v1/application_templates/{template_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
template_id	string	ポリシーテンプレートの固有識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
name	string	（オプション）インポート時にテンプレートの名前として使用されます。

属性	タイプ	説明
説明	string	（オプション）適用プロセス中に表示されるテンプレートの説明

応答オブジェクト：指定されたIDの変更されたポリシーテンプレートオブジェクトを返します。

サンプル python コード

```
new_name = <new-name>
payload = {'name': new_name}
template_id = '<template-id>'
restclient.post('/application_templates/%s' % template_id,
                json_body=json.dumps(payload))
```

ポリシーテンプレートの削除

このエンドポイントは、指定されたポリシーテンプレートを削除します。

```
DELETE /openapi/v1/application_templates/{template_id}
```

パラメータ：要求URLには、次のパラメータが含まれています。

名前	タイプ	説明
template_id	string	ポリシーテンプレートの固有識別子。

応答オブジェクト：なし

サンプル python コード

```
template_id = '<template-id>'
restclient.delete('/application_templates/%s' % template_id)
```

ポリシーテンプレートのダウンロード

このエンドポイントは、ポリシーテンプレートをダウンロードします。

```
GET /openapi/v1/application_templates/{template_id}/download
```

パラメータ：要求URLには、次のパラメータが含まれています。

名前	タイプ	説明
template_id	string	ポリシーテンプレートの固有識別子。

応答オブジェクト：指定されたIDを持つ完全なポリシーテンプレート定義を返します。

サンプル python コード

```
template_id = '<template-id>'
restclient.get('/application_templates/%s/download' % template_id)
```

クラスタ

この一連のAPIを使用して、ワークスペース（「アプリケーション」）のメンバーであるクラスタを追加、編集、または削除できます。APIキーに関連付けられた`user_role_scope_management`機能が必要です。

クラスタオブジェクト

クラスタオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	クラスタの一意の識別子。
consistent_uuid	string	自動ポリシー検出実行時の一貫したID。
application_id	string	クラスタが属するワークスペースのID。
version	string	クラスタが属するワークスペースのバージョン
name	string	クラスタの名前。
説明	string	クラスタの説明。
approved	boolean	クラスタがユーザーによって「承認」されている場合。
query	JSON	親範囲のフィルタと組み合わせて、フィルタに関連付けられているフィルタ(または一致基準)。
short_query{1}JSON{1}	JSON	フィルタに関連付けられているフィルタ(または一致基準)。
alternate_queries	クエリの配列	動的モードで実行される自動ポリシー検出によって生成された代替の推奨クエリ。
インベントリ	インベントリの配列	要求された場合、IP、ホスト名、vrf_id、およびuuidを含むクラスタのメンバーインベントリを返します。

クラスタの取得

このエンドポイントは、特定のワークスペース（「application」）のクラスタのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/clusters
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	クラスタが属するワークスペースの ID。
version	string	クラスタを取得するワークスペースのバージョンの表示。
include_inventory	boolean	クラスタのインベントリを含みます。

応答オブジェクト：この特定のワークスペースとバージョンの全クラスタの配列を返します。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
restclient.get('/applications/%s/clusters' % application_id)
```

特定のクラスタの取得

このエンドポイントは、クラスタのインスタンスを返します。

```
PUT /openapi/v1/clusters/{cluster_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
cluster_id	string	クラスタの一意の識別子。
include_inventory	boolean	クラスタのインベントリを含みます。

応答オブジェクト：指定された ID に関連付けられたクラスタオブジェクトを返します。

サンプル python コード

```
cluster_id = '5d02d021497d4f0949ba74e4'
restclient.get('/clusters/%s' % cluster_id)
```

クラスタの作成

このエンドポイントは、新しいクラスタを作成するために使用されます。

```
POST /openapi/v1/applications/{application_id}/clusters
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	クラスタが属するワークスペースの ID。

JSON クエリの本文には、次のキーが含まれています。

属性	タイプ	説明
name	string	クラスタの名前。
version	string	クラスタが追加されるワークスペースのバージョンを示します。
説明	string	（オプション）クラスタの説明。
approved	boolean	（オプション）承認されたクラスタは、自動ポリシー検出の実行中に更新されません。デフォルトは「False」です。
query	JSON	フィルタに関連付けられているフィルタ(または一致基準)。代替クエリモード（動的モードとも呼ばれます）は、ワークスペースで有効にする必要があります。それ以外の場合は無視されます。
query	JSON	フィルタに関連付けられているフィルタ(または一致基準)。代替クエリモード（動的モードとも呼ばれます）は、ワークスペースで有効にする必要があります。それ以外の場合は無視されます。
ノード	配列	IP アドレスまたはエンドポイントのリスト。クエリが提供済みでワークスペースが動的モードでない限り、これらの IP に一致するクエリを作成するために使用されます。

ノード オブジェクトの属性：

名前	タイプ	説明
ip	string	IP アドレス
name	string	（オプション）ノードの名前。
prefix_len	integer	（オプション）サブネットマスク。

（注） クエリが提供済みでワークスペースが動的モードでない限り、ノードはクエリの作成に使用されます。

応答オブジェクト：新しく作成されたクラスタオブジェクトを返します。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
payload = {
  'name': 'test_cluster',
  'version': 'v2',
  'description': 'basic granularity',
  'approved': False,
  'query': {
      'type': 'eq',
      'field': 'host_name',
      'value': 'centos6001'
  }
}
restclient.post('/applications/%s/clusters' % application_id)
```

クラスタの更新

このエンドポイントは、クラスタを更新します。

```
PUT /openapi/v1/clusters/{cluster_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
cluster_id	string	クラスタの一意の識別子。

JSON クエリの本文には、次のキーが含まれています。

属性	タイプ	説明
name	string	クラスタの名前。

属性	タイプ	説明
説明	string	（オプション）クラスタの説明。
approved	boolean	承認されたクラスタは、自動ポリシー検出の実行中に更新されません。
query	JSON	フィルタに関連付けられているフィルタ(または一致基準)。代替クエリモード（動的モードとも呼ばれます）は、ワークスペースで有効にする必要があります。それ以外の場合は無視されます。

応答オブジェクト：指定された ID に関連付けられている変更されたクラスタオブジェクトを返します。

サンプル python コード

```
cluster_id = '5d02d2a4497d4f5194f104ef'
payload = {
  'name': 'new_test_cluster',
}
restclient.put('/clusters/%s' % cluster_id, json_body=json.dumps(payload))
```

クラスタの削除

このエンドポイントは、指定されたクラスタを削除します。クラスタがポリシーによって使用されている場合、クラスタは削除されず、依存関係のリストが返されます。

```
DELETE /openapi/v1/clusters/{cluster_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
cluster_id	string	クラスタの一意の識別子。

応答オブジェクト：なし

サンプル python コード

```
cluster_id = '5d02d2a4497d4f5194f104ef'
restclient.delete('/clusters/%s' % cluster_id)
```

カンバセーション

カンバセーションは、コンシューマポートが削除された自動ポリシー検出の実行における時間範囲内の集約されたフローです。カンバセーションに関する詳細な説明は、「カンバセーション」を参照してください。

このAPIを使用すると、特定のワークスペースに対する自動ポリシー検出の実行中に生成されたカンバセーションを検索できます。このAPIには、APIキーに関連付けられている`app_policy_management`機能が必要です。

ポリシー検出実行での会話の検索

このエンドポイントを使用すると、特定のワークスペースに対する自動ポリシー検出の実行中の会話を検索できます。ダウンロードした会話の一部として表示させる、サポートされているディメンションとメトリックのサブセットを指定することもできます。必要に応じて、サポートされているディメンションとメトリックに関するフィルタを使用して、会話のサブセットに対するクエリを実行できます。

```
POST /openapi/v1/conversations/{application_id}
```

クエリは、次のキーを使用したJSON本文で構成されます。

名前	タイプ	説明
version	integer	自動ポリシー検出実行のバージョン
filter	JSON	（オプション）クエリフィルタ。フィルタが空（すなわち{}）の場合、クエリはすべての会話に一致します。サポートされているディメンションとメトリックのフィルタを使用して、より具体的な会話をダウンロードできます。フィルタの構文については、フィルタを参照してください。
寸法	array	（オプション）ダウンロードした会話に対して返されるディメンションのリスト。サポートされているディメンションのリストは、サポートされているディメンションにあります。

名前	タイプ	説明
メトリック	array	（オプション）ダウンロードした会話に対して返されるメトリックのリスト。サポートされているメトリックのリストは、サポートされているメトリック にあります。
limit	integer	（オプション）1 つの API 応答で返される会話の数。
offset	string	（オプション）前の応答から受信したオフセット（ページネーションに有用）。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
    "version": 1,
    "filter": {
        "type": "and",
        "filters":[
            {
                "type": "eq",
                "field": "excluded",
                "value": False
            },
            {
                "type": "eq",
                "field": "protocol",
                "value": "TCP"
            },
        ]
    },
    "dimensions": ["src_ip", "dst_ip", "port"],
    "metrics": ["byte_count", "packet_count"],
    "limit" : 2,
    "offset": <offset-object>
}
```

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

キー	値
offset	結果の次のページでパスされる応答オフセット
results	結果のリスト

結果の次のページを生成するには、`offset` の応答で受信したオブジェクトを取得し、それを次のクエリの `offset` の値として渡します。

```
    req_payload = {"version": 1,
                   "limit": 10,
                   "filter": {"type": "and",
                       "filters": [
                           {"type": "eq", "field": "excluded", "value": False},
                           {"type": "eq", "field": "protocol", "value": "TCP"}
                        ]
                   }
    }

    resp = restclient.post('/conversations/{application_id}',
json_body=json.dumps(req_payload))
    print resp.status_code
    if resp.status_code == 200:
        parsed_resp = json.loads(resp.content)
        print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

ポリシー検出実行の上位 N 件の会話

このエンドポイントを使用すると、メトリックに基づき、ディメンションでグループ化された、特定のワークスペースに対して実行された自動ポリシー検出の上位の会話を検索できます。現在サポートされているメトリックはサポートされているメトリック、ディメンションで現在サポートされているグループはサポートされているディメンションです。サポートされているディメンションとメトリックのフィルタを使用して、会話のサブセットをクエリできます。たとえば、バイトトラフィックの会話が最も多いソース IP アドレスを検索するには、`src_ip` ディメンションと `byte_count` メトリックを指定してクエリを使用します。

```
POST /openapi/v1/conversations/{application_id}/topn
```

クエリは、次のキーを使用した JSON 本文で構成されます。

名前	タイプ	説明
version	整数	自動ポリシー検出実行のバージョン
dimension	string	上位 N 個のクエリに対してグループ化される会話のディメンション。 サポートされているディメンション：src_ip、dst_ip
metric	string	上位 N 件の会話に対して並べ替えられるメトリック。サポートされているメトリックのリストは、 サポートされているメトリック にあります。

名前	タイプ	説明
filter	JSON	（オプション）クエリフィルタ。フィルタが空（つまり、{}）の場合、クエリはすべての会話に一致します。サポートされているディメンションとメトリックのフィルタを使用して、より具体的な会話をダウンロードできます。フィルタの構文については、フィルタ を参照してください。
しきい値	integer	単一の API 応答で返される上位 N 件の結果の数。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
    "version": 1,
    "dimension": "src_ip",
    "metric": "byte_count",
    "filter": {
        "type": "and",
        "filters":[
            {
                "type": "eq",
                "field": "excluded",
                "value": False
            },
            {
                "type": "eq",
                "field": "protocol",
                "value": "TCP"
            },
        ]
    },
    "threshold" : 10
}
```

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

キー	値
results	結果キーを持つ1つのJSONオブジェクトと、クエリディメンションとメトリックに一致するキーを持つ結果オブジェクトのリストの値をリストします。

```
[ {"result": [
  {
```

```
      "byte_count": 1795195565,
      "src_ip": "192.168.1.6"
    },
    {
      "byte_count": 1781002379,
      "src_ip": "192.168.1.28"
    },
    ...
  ] } ]


    req_payload = {"version": 1, "dimension": "src_ip", "metric": "byte_count",
      "filter": {"type": "and",
        "filters": [
          {"type": "eq", "field": "excluded", "value": False},
          {"type": "eq", "field": "protocol", "value": "TCP"},
          {"type": "eq", "field": "consumer_filter_id", "value":
"16b12a5614c5af5b68afa7ce"},
          {"type": "subnet", "field": "src_ip", "value": "192.168.1.0/24"}
        ]
      },
      "threshold" : 10
    }

    resp = restclient.post('/conversations/{application_id}/topn',
json_body=json.dumps(req_payload))
    print resp.status_code
    if resp.status_code == 200:
        parsed_resp = json.loads(resp.content)
        print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

サポートされているディメンション

名前	タイプ	説明
src_ip	string	コンシューマの IP アドレス
dst_ip	string	コンシューマの IP アドレス
protocol	string	通信で使用されたプロトコル。例：「TCP」、「UDP」など
port	integer	プロバイダーのポート。
address_type	string	「IPv4」または「IPv6」
consumer_filter_id	string	コンシューマ IP がクラスタに属する場合はクラスタのクラスタ ID。それ以外の場合はコンシューマ IP が属する範囲 ID。

名前	タイプ	説明
provider_filter_id	string	プロバイダー IP がクラスタに属している場合はクラスタのクラスタ ID、それ以外の場合はプロバイダー IP が属する範囲 ID。
excluded	boolean	ポリシーの生成中にこの会話を除外するかどうか。
confidence	double	コンシューマとプロバイダーの分類の信頼度。値は 0.0 から 1.0 まで変化し、1.0 は分類について高い信頼度を意味します。

サポートされているメトリック

名前	タイプ	説明
byte-count	integer	会話中のバイトの総数
packet_count	integer	会話中のパケットの総数

除外フィルタ

この一連の API は、除外フィルタを追加、編集、または削除するために使用でき、API キーに関連付けられている `user_role_scope_management` 機能が必要です。

除外フィルタは、自動ポリシー検出クラスタリングアルゴリズムからフローを除外します。詳細については、「除外フィルタ」を参照してください。

除外フィルタオブジェクト

除外フィルタオブジェクトの属性について、以下で説明します。

属性	タイプ	説明
id	string	クラスタの一意の識別子。
application_id	string	除外フィルタが属するワークスペースの ID。
version	string	除外フィルタが属するワークスペースのバージョン。

属性	タイプ	説明
consumer_filter_id	string	定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのコンシューマとして使用できます。
provider_filter_id	string	定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのプロバイダーとして使用できます。
proto	integer	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
port	array	ポートの包含範囲。例：[80, 80] または [5000, 6000]。NULL はすべてのポートを意味します。
updated_at	integer	除外フィルタが更新されたときの UNIX タイムスタンプ。

除外フィルタの取得

このエンドポイントは、特定のワークスペースの除外フィルタのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/exclusion_filters
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
version	string	除外フィルタを取得するワークスペースのバージョンを示します。

応答オブジェクト：指定されたワークスペースとバージョンの除外フィルタオブジェクトのリストを返します。

サンプル python コード

```
application_id = '<application-id>'
params = {'version': 'v10'}
restclient.get('/applications/%s/exclusion_filters' % application_id,
               params=params)
```

特定の除外フィルタを取得する

このエンドポイントは、除外フィルタのインスタンスを返します。

```
GET /openapi/v1/exclusion_filters/{exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
exclusion_filter_id	string	除外フィルタの一意の識別子。

応答オブジェクト：指定された ID を持つ除外フィルタオブジェクトを返します。

サンプル python コード

```
exclusion_filter_id = '<exclusion-filter-id>'
restclient.get('/exclusion_filters/%s' % exclusion_filter_id)
```

除外フィルタの作成

このエンドポイントは、除外フィルタを作成するために使用します。

```
POST /openapi/v1/applications/{application_id}/exclusion_filters
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
version	string	除外フィルタが属するワークスペースのバージョン。
consumer_filter_id	string	（オプション）定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのコンシューマとして使用できます。

属性	タイプ	説明
provider_filter_id	string	（オプション）定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのプロバイダーとして使用できます。
proto	整数	（オプション）プロトコル整数値（NULL はすべてのプロトコルを意味します）。
start_port	integer	（オプション）範囲の開始ポート。
end_port	integer	（オプション）範囲の終了ポート。

省略可能なパラメータが欠落している場合は、ワイルドカード（いずれにも一致）と見なされます。

応答オブジェクト：作成された除外フィルタオブジェクトを返します。

サンプル python コード

```
provider_filter_id = '<provider-filter-id>'
consumer_filter_id = '<consumer-filter-id>'
payload = {'version': 'v0',
           'consumer_filter_id': consumer_filter_id,
           'provider_filter_id': provider_filter_id,
           'proto': 6,
           'start_port': 800,
           'end_port': 1000}
application_id = '<application-id>'
restclient.post('/applications/%s/exclusion_filters' % application_id,
                json_body=json.dumps(payload))
```

除外フィルタを更新する

このエンドポイントは除外フィルタを更新します。

```
PUT /openapi/v1/exclusion_filters/{exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
exclusion_filter_id	string	除外フィルタの一意の識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
consumer_filter_id	string	（オプション）定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのコンシューマとして使用できます。
provider_filter_id	string	（オプション）定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのプロバイダーとして使用できます。
proto	integer	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
start_port	integer	（オプション）範囲の開始ポート。
end_port	integer	（オプション）範囲の終了ポート。

応答オブジェクト：指定された ID を持つ変更された除外フィルタオブジェクトを返します。

サンプル python コード

```
payload = {'proto': 17}
exclusion_filter_id = '<exclusion-filter-id>'
restclient.post('/exclusion_filters/%s' % exclusion_filter_id,
                json_body=json.dumps(payload))
```

除外フィルタの削除

このエンドポイントは、指定された除外フィルタを削除します。

```
DELETE /openapi/v1/exclusion_filters/{exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
exclusion_filter_id	string	除外フィルタの一意の識別子。

応答オブジェクト：なし

サンプル python コード

```
exclusion_filter_id = '<exclusion-filter-id>'
restclient.delete('/exclusion_filters/%s' % exclusion_filter_id)
```

デフォルトの除外フィルタ

この一連の API は、除外フィルタを追加、編集、または削除するために使用できます。API キーに関連付けられている `app_policy_management` 機能が必要です。

除外フィルタは、自動ポリシー検出クラスタリングアルゴリズムからフローを除外します。詳細については、「除外フィルタ」を参照してください。

デフォルト除外フィルタのオブジェクト

除外フィルタオブジェクトの属性について、以下で説明します。

属性	タイプ	説明
id	string	デフォルト除外フィルタの一意の識別子。
consumer_filter_id	string	定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのコンシューマとして使用できます。
provider_filter_id	string	定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのプロバイダーとして使用できます。
proto	integer	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
port	array	ポートの包含範囲。例：[80, 80] または [5000, 6000]。NULL はすべてのポートを意味します。
updated_at	integer	除外フィルタが更新されたときの UNIX タイムスタンプ。

デフォルトの除外フィルタの取得

このエンドポイントは、デフォルトの除外フィルタのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/default_exclusion_filters?root_app_scope_id={root_app_scope_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_app_scope_id	string	ルート範囲の一意の識別子。

応答オブジェクト：ルート範囲の既定の除外フィルタオブジェクトのリストを返します。

サンプル python コード

```
root_app_scope_id = '<root-app-scope-id>'
 restclient.get('/default_exclusion_filters?root_app_scope_id=%s' % root_app_scope_id)
```

特定のデフォルト除外フィルタを取得

このエンドポイントは、デフォルト除外フィルタのインスタンスを返します。

```
default_exclusion_filter_id = '<default-exclusion-filter-id>'

restclient.get('/default_exclusion_filters/%s' % default_exclusion_filter_id)
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
default_exclusion_filter_id	string	除外フィルタの一意の識別子。

応答オブジェクト：指定されたIDを持つデフォルトの除外フィルタオブジェクトを返します。

サンプル python コード

```
default_exclusion_filter_id = '<default-exclusion-filter-id>'
restclient.get('/default_exclusion_filters/%s' % default_exclusion_filter_id)
```

デフォルトの除外フィルタの作成

このエンドポイントは、デフォルトの除外フィルタを作成するために使用します。

```
GET /openapi/v1/default_exclusion_filters?root_app_scope_id={root_app_scope_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_app_scope_id	string	ルート範囲の一意の識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
consumer_filter_id	string	（オプション）定義された範囲またはインベントリフィルタの ID。

属性	タイプ	説明
provider_filter_id	string	（オプション）定義された範囲またはインベントリフィルタの ID。
proto	整数	（オプション）プロトコル整数値（NULL はすべてのプロトコルを意味します）。
start_port	integer	（オプション）範囲の開始ポート。
end_port	integer	（オプション）範囲の終了ポート。

応答オブジェクト：作成されたデフォルトの除外フィルタオブジェクトを返します。

サンプル python コード

```
provider_filter_id = '<provider-filter-id>'
consumer_filter_id = '<consumer-filter-id>'
payload = {'consumer_filter_id': consumer_filter_id,
           'provider_filter_id': provider_filter_id,
           'proto': 6,
           'start_port': 800,
           'end_port': 1000}
root_app_scope_id = '<root-app-scope-id>'
restclient.post('/default_exclusion_filters?root_app_scope_id=%s' % root_app_scope_id,

                json_body=json.dumps(payload))
```

デフォルトの除外フィルタの更新

このエンドポイントは、デフォルトの除外フィルタを更新します。

```
PUT /openapi/v1/default_exclusion_filters/{default_exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
default_exclusion_filter_id	string	デフォルトの除外フィルタの一意の識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
consumer_filter_id	string	（オプション）定義された範囲またはインベントリフィルタの ID。
provider_filter_id	string	（オプション）定義された範囲またはインベントリフィルタの ID。
proto	integer	Protocol Integer value (NULL はすべてのプロトコルを意味します)。

start_port	integer	（オプション）範囲の開始ポート。
end_port	integer	（オプション）範囲の終了ポート。

応答オブジェクト：指定された ID を持つ変更されたデフォルトの除外フィルタオブジェクトを返します。

サンプル python コード

```
payload = {'proto': 17}
default_exclusion_filter_id = '<default-exclusion-filter-id>'
restclient.post('/default_exclusion_filters/%s' % default_exclusion_filter_id,
                json_body=json.dumps(payload))
```

デフォルトの除外フィルタの削除

このエンドポイントは、指定されたデフォルトの除外フィルタを削除します。

```
DELETE /openapi/v1/default_exclusion_filters/{default_exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
default_exclusion_filter_id	string	除外フィルタの一意の識別子。

応答オブジェクト：なし

サンプル python コード

```
default_exclusion_filter_id = '<default-exclusion-filter-id>'
restclient.delete('/default_exclusion_filters/%s' % default_exclusion_filter_id)
```

ライブ分析

ライブ分析やポリシー分析は、セキュリティポリシーの生成における重要な側面です。これにより、一連のポリシー（自動ポリシー検出によって生成された場合、またはユーザーによって手動で追加された場合）を評価してから、実際にそれらのポリシーをワークロードに適用できます。ライブ分析を使用すると、アプリケーショントラフィックを中断することなく、ライブトラフィックで what-if 分析を実行できます。

このセクションに記載されている一連の API を使用すると、フローをダウンロードし、ワークスペースで現在公開されているの一連のポリシーがそれらのフローに及ぼす影響を確認できます。この一連の API を起動するには、API キーに関連付けられている `app_policy_management` 機能が必要です。

ライブ分析を介して利用できるフローには属性(ディメンションとメトリック)があります。ダウンロード API を使用すると、ユーザーはディメンションのさまざまな基準でフローをフィルタリングできます。

ライブ分析でのフローサイズ

このエンドポイントは、ライブ分析を介して利用可能なフローをダウンロードするために、検索条件（またはフィルタ）を指定できる列を把握するために役立ちます。最も一般的なユースケースは、許可されたか、エスケープされたか、または拒否されたフローをダウンロードすることです。カテゴリサイズの検索条件をダウンロードAPIに渡すことでダウンロードが実現します。[type: eq] とともに使用する場合、フローのインバウンドカテゴリおよびアウトバウンドカテゴリが一致している必要があります。[type: contains] とともに使用する場合、フローのインバウンドカテゴリおよびアウトバウンドカテゴリが一致している必要があります。

```
GET /openapi/v1/live_analysis/dimensions
```

ライブ分析で使用可能なフローメトリック

このエンドポイントは、ライブ分析に関連付けられたメトリック（バイト数、パケット数など）のリストを返します。このエンドポイントの 1 つの使用例は、ダウンロード API にメトリックのサブセットを含めることです。つまり、すべてのメトリックをダウンロードする代わりに、ユーザーは関心のあるメトリックの小さなサブセットを指定できます。

```
GET /openapi/v1/live_analysis/metrics
```

ライブ分析を介して利用可能なフローのダウンロード

このエンドポイントは、フィルタ条件に一致するフローのリストを返します。結果に含まれる各フローオブジェクトには、ライブ分析ディメンション（上記のライブ分析ディメンション API によって返される）とライブ分析メトリック（上記のライブ分析メトリック API によって返される）の結合である属性があります。使用可能なディメンションとメトリックの完全なセットに関心がない場合ユーザーは、必要に応じて、ディメンションまたはメトリックの小さなサブセットを指定することもできます。ディメンションまたはメトリックのより小さなサブセットの予測には、API 呼び出しが高速になるという副次的効果もあります。

```
POST /openapi/v1/live_analysis/{application_id}
```

クエリ本文は、次のキーを使用した JSON 本文で構成されます。

名前	タイプ	説明
t0	整数または文字列	時間間隔の開始時刻（エポックまたは ISO 8601）
t1	整数または文字列	時間間隔の終了時刻（エポックまたは ISO 8601）
filter	JSON	クエリ フィルタ。フィルタが空 (例: {}) の場合、クエリはすべてのフローに一致します。フィルタの構文については、フロー検索の「フィルタ」セクションを参照してください。

名前	タイプ	説明
寸法	array	（オプション）ライブ分析を介して利用可能なダウンロードされたフローに対して返されるフローディメンションのリスト。指定しない場合、利用可能なすべてのディメンションが返されます。
メトリック	array	（オプション）ライブ分析を介して利用可能なダウンロードされたフローに対して返されるフローメトリックのリスト。
limit	integer	（オプション）単一の API 応答で返される フローの数。
offset	string	（オプション）前の応答から受信したオフセット（ページネーションに有用）。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
    "t0": "2016-06-17T09:00:00-0700",
    "t1": "2016-06-17T17:00:00-0700",
    "filter": {
        "type": "and",
        "filters": [
            {
                "type": "contains",
                "field": "category",
                "value": "escaped"
            },
            {
                "type": "in",
                "field": "dst_port",
                "values": ["80", "443"]
            }
        ]
    },
    "limit": 100,
    "offset": <offset-object>
}
```

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

キー	値
offset	結果の次のページでパスされる応答オフセット
results	結果のリスト

結果の次のページを生成するには、`offset` の応答で受信したオブジェクトを取得し、それを次のクエリの `offset` の値として渡します。

サンプル python コード

```
   req_payload = {"t0": "2016-11-07T09:00:00-0700",
                  "t1": "2016-11-07T19:00:00-0700",
                  "limit": 10,
                  "filter": {"type": "and",
                      "filters": [
                         {"type": "contains", "field": "category", "value": "escaped"},

                          {"type": "regex", "field": "src_hostname", "value": "web*"}
                       ]
                  }
    }

    resp = restclient.post('/live_analysis/{application_id}',
json_body=json.dumps(req_payload))
    print resp.status_code
    if resp.status_code == 200:
        parsed_resp = json.loads(resp.content)
        print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

スコープ

この一連の API を使用して、Secure Workload クラスタ展開の範囲（または AppScopes）を管理できます。API キーに関連付けられた `user_role_scope_management` 機能が必要です。範囲リストを取得する API は、 `app_policy_management` または `sensor_management` の機能を持つ API キーで使用できます。

範囲オブジェクト

範囲オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	範囲の固有識別子。
short_name	string	ユーザーが指定した範囲の名前。

属性	タイプ	説明
name	string	範囲の完全修飾名。これは完全修飾名です。つまり、ルート範囲に至るまで、親範囲の名前 (該当する場合) があります。
説明	string	ユーザーが指定した範囲の説明。
short_query{1}JSON{1}	JSON	範囲に関連付けられているフィルタ (または一致基準)。
query	JSON	（ルート範囲までの）親範囲のフィルタとともに範囲に関連付けられるフィルタ（一致基準）。
vrf-id	整数	範囲が属している VRF の ID。
parent_app_scope_id	string	親範囲の ID。
child_app_scope_ids	array	範囲の子 id の配列。
policy_priority		ワークスペースの優先順位を並べ替えるために使用されます。「ポリシー属性」を参照してください。
ダーティ	bool	子または親クエリが更新され、変更をコミットする必要があることを示します。
dirty_short_query	JSON	この範囲のクエリが更新されていてもコミットされていない場合は、null 以外です。

範囲の取得

このエンドポイントは、Secure Workload アプライアンスが認識している範囲のリストを返します。この API は、 `app_policy_management` または `user_role_scope_management` のいずれかの機能を持つ API キーで使用できます。

```
GET/openapi/v1/app_scopes
```

パラメータ：

名前	タイプ	説明
vrf-id	integer	アプリの範囲を vrf_id で照合します。
root_app_scope_id	string	アプリの範囲をルートアプリの範囲 ID と照合します。
exact_name	string	大文字と小文字を区別して、正確な名前に一致する範囲を返します。
exact_short_name	string	大文字と小文字を区別して、正確な short_name に一致する範囲を返します。

範囲オブジェクトのリストを戻します。

範囲の作成

このエンドポイントは、新しい範囲を作成するために使用されます。

`POST/openapi/v1/app_scopes`

パラメータ：

名前	タイプ	説明
short_name	string	ユーザーが指定した範囲の名前。
説明	string	ユーザーが指定した範囲の説明。
short_query{1}JSON{1}	JSON	範囲に関連付けられているフィルタ (または一致基準)。
parent_app_scope_id	string	親範囲の ID。
policy_priority	integer	デフォルトは「last」です。ワークスペースの優先順位を並べ替えるために使用されます。「自動検出されたポリシーの確認」の「ポリシーの順序付け」を参照してください。

サンプル python コード

```
    req_payload = {"t0": "2016-11-07T09:00:00-0700",
                   "t1": "2016-11-07T19:00:00-0700",
                   "limit": 10,
                   "filter": {"type": "and",
                       "filters": [
                          {"type": "contains", "field": "category", "value": "escaped"},

                           {"type": "regex", "field": "src_hostname", "value": "web*"}
                        ]
                   }
    }

    resp = restclient.post('/live_analysis/{application_id}',
json_body=json.dumps(req_payload))
    print resp.status_code
    if resp.status_code == 200:
        parsed_resp = json.loads(resp.content)
        print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

サブネットに基づいて範囲を作成するには、次の short_query を使用します。

```
    "short_query":
      {
        "type":"subnet",
        "field":"ip",
        "value": "1.0.0.0/8"
      },
```

特定の範囲を取得する

このエンドポイントは、範囲のインスタンスを返します。

`GET/openapi/v1/app_scopes/{app_scope_id}`

指定した ID に関連付けられている範囲オブジェクトを返します。

範囲の更新

このエンドポイントは範囲を更新します。`name` および `description` への変更はすぐに適用されます。short_query への変更は、範囲を「ダーティ」としてマークし、`dirty_short_query` 属性を設定します。特定のルート範囲の下にあるすべての範囲クエリが変更になった場合、範囲クエリの変更のコミットエンドポイントに ping を実行して、必要なすべての更新をコミットする必要があります。

`PUT/openapi/v1/app_scopes/{app_scope_id}`

パラメータ：

名前	タイプ	説明
short_name	string	ユーザーが指定した範囲の名前。
説明	string	ユーザーが指定した範囲の説明。

名前	タイプ	説明
short_query{1}JSON{1}	JSON	範囲に関連付けられているフィルタ (または一致基準)。

指定された ID に関連付けられている変更された範囲オブジェクトを返します。

特定範囲の削除

このエンドポイントは、指定された範囲を削除します。

```
DELETE /openapi/v1/app_scopes/{app_scope_id}
```

範囲がワークスペース、ポリシー、ユーザー インベントリ フィルタなどに関連付けられている場合、このエンドポイントは `422 Unprocessable Entity` を返します。返されるエラーオブジェクトには、`details` 属性が含まれ、依存オブジェクトの数とともに各タイプの最初の 10 個の ID が示されます。この情報を使用して、問題となっている依存関係を見つけて削除できます。

ポリシーの優先順位で範囲を取得する

このエンドポイントは、対応するプライマリワークスペースが適用される順序で範囲を一覧表示します。

```
GET /openapi/v1/app_scopes/{root_app_scope_id}/policy_order
```

範囲オブジェクトの配列データを返します。

ポリシー順序の更新

このエンドポイントは、ポリシーが適用される順序を更新します。



警告 このエンドポイントは、ポリシーが適用される順序を変更します。その結果、新しいホストファイアウォール ルールが挿入され、関連するホスト上で既存のルールがすべて削除されます。

```
POST /openapi/v1/app_scopes/{root_app_scope_id}/policy_order
```

パラメータ：

名前	タイプ	説明
root_app_scope_id	string	順序が変更されるルート範囲。
ids	アレイ	適用される順序での範囲 ID 文字列の配列。

ids 配列パラメータには、ルート範囲のすべてのメンバー（ルートを含む）が含まれている必要があります。

範囲クエリの変更のコミット

このエンドポイントは非同期バックグラウンドジョブをトリガーして、特定のルート範囲のすべての「ダーティ」子を更新します。このジョブは、範囲とワークスペースを更新します。詳細については、「スコープ」を参照してください。

```
POST/openapi/v1/app_scopes/commit_dirty
```

パラメータ：

名前	タイプ	説明
root_app_scope_id	string	すべての子が更新されるルート範囲の ID。
sync	boolean	（オプション）リクエストを同期する必要があるかどうかを示します。

ジョブがキューに入れられたことを示すために、202を 返します。ジョブが完了したかどうかを確認するには、ルート範囲の「ダーティ」属性をポーリングして、それが false に設定されているかどうかを確認します。

ユーザーは、sync パラメータを指定して、ジョブをすぐに実行することもできます。リクエストが完了すると、ステータスコード 200 が返されます。多くの更新を適用する必要がある場合、このリクエストには時間がかかることがあります。

グループ提案リクエストの送信

範囲のグループ提案リクエストを送信します。

```
PUT /openapi/v1/app_scopes/{app_scope_id}/suggest_groups
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
app_scope_id	string	範囲の固有識別子。

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
start_time	string	グループ提案入力時間間隔の開始時間。
end_time	string	グループ提案入力時間間隔の終了時間。

応答オブジェクト：次の属性を持つオブジェクトを返します。

名前	タイプ	説明
message	string	グループ提案リクエストの提出成否に関するメッセージです。

サンプル python コード

```
app_scope_id = '5d02b493755f0237a3d6e078'
req_payload = {
      'start_time': '2020-09-17T10:00:00-0700',
      'end_time': '2020-09-17T11:00:00-0700',
}
resp = restclient.put('/app_scopes/%s/suggest_groups' % app_scope_id,
                      json_body=json.dumps(req_payload))
```

グループ提案ステータスの取得

範囲のグループ提案ステータスのクエリを実行します。

```
GET /openapi/v1/app_scopes/{app_scope_id}/suggest_groups_status
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
app_scope_id	string	範囲の固有識別子。

応答オブジェクト：次の属性を持つオブジェクトを返します。

名前	タイプ	説明
status	string	グループ提案のステータス。値：PENDING、COMPLETE、または FAILED

サンプル python コード

```
app_scope_id = '5d02b493755f0237a3d6e078'
resp = restclient.get('/app_scopes/%s/suggest_groups_status' % app_scope_id)
```

アラート

この一連の API を使用して、ユーザーアラートを管理できます。API キーに関連付けられた `user_alert_management` 機能が必要です。

- アラートオブジェクト （1128 ページ）
- アラートの取得 （1128 ページ）
- アラートの作成 （1129 ページ）

• 特定のアラートの取得 （1130 ページ）
• アラートの更新 （1130 ページ）
• 特定のアラートの削除 （1130 ページ）

アラートオブジェクト

各アラート設定オブジェクトには、次のフィールドが含まれています。

属性	タイプ	説明
app_name	string	アラート設定に関連付けられているアプリケーション名。
ルール	オブジェクト	アラート設定がアラートをトリガーするために満たされる必要がある一連の条件。
subjects	オブジェクト	アラートを受信するユーザーのリスト。
severity	string	アラート設定に関連付けられている重大度のレベルを示します。
individual_alert	boolean	アラート設定をトリガーする個々のアラートを送信するかどうかを示します。
summary_alert_freq	string	特定のアラート設定に対して送信されるサマリーアラートの頻度。
alert_type	string	アラート設定の固有識別子。
app_instance_id	string	アラート設定に関連付けられている特定のインスタンスの固有識別子。

アラートの取得

このエンドポイントは、ユーザーのアラート設定のリストを取得します。アラートは、指定されたルート範囲にフィルタリングできます。範囲が指定されていない場合、ユーザーがアクセスできるすべての範囲のすべてのアラートが返されます。サービスプロバイダーアラートは、ユーザーがサイト管理者である場合にのみ返されます。

```
GET /openapi/v1/alert_confs
```

パラメータ：

名前	タイプ	説明
root_app_scope_id	string	（オプション）その範囲に割り当てられているアラートのみを返すルート範囲の ID。

応答オブジェクト：ユーザー アラート オブジェクトのリストを返します。

アラートの作成

このエンドポイントは、新しいアラートを作成するために使用されます。

```
POST openapi/v1/alert_confs
```

パラメータ：

属性	タイプ	説明
app_name	string	アラート設定に関連付けられているアプリケーション名。
ルール	オブジェクト	アラート設定がアラートをトリガーするために満たされる必要がある一連の条件。
subjects	オブジェクト	アラートを受信するユーザーのリスト。
severity	文字列	アラート設定に関連付けられている重大度のレベルを示します。
individual_alert	boolean	アラート設定をトリガーする個々のアラートを送信するかどうかを示します。
summary_alert_freq	string	特定のアラート設定に対して送信されるサマリーアラートの頻度。
alert_type	string	アラート設定の固有識別子。
app_instance_id	string	アラート設定に関連付けられている特定のインスタンスの固有識別子。

要求側のユーザーは、指定された範囲にアクセスできる必要があります。範囲のないアラートは「サービス プロバイダー アラート」であり、サイト管理者のみが作成できます。

応答オブジェクト：新しく作成されたアラートオブジェクトを返します。

特定のアラートの取得

このエンドポイントは、特定のアラートオブジェクトを返します。

```
GET /openapi/v1/alert_confs/
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
alert_id	string	アラートを一意に識別します。

応答オブジェクト：指定された ID に関連付けられているアラートオブジェクトを返します。

アラートの更新

このエンドポイントは、既存のアラートを更新するために使用されます。

```
PUT /openapi/v1/alert_confs/
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
alert_id	string	このアラートの設定を取得または変更します。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
name	string	ユーザーが指定したアラートの名前。
説明	string	ユーザーが指定したアラートの説明。

要求側のユーザーは、指定された範囲にアクセスできる必要があります。範囲のないアラートは「サービス プロバイダー アラート」と呼ばれ、サイト管理者のみが更新できます。

応答オブジェクト：指定された ID を持つ更新されたアラートオブジェクト。

特定のアラートの削除

このエンドポイントは、指定されたアラートを削除します。

```
DELETE /openapi/v1/alert_confs/{alert_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
alert_id	string	アラートを一意に識別します。

応答オブジェクト：なし。

ロール

この一連の API を使用して、ユーザー ロールを管理できます。API キーに関連付けられた `user_role_scope_management` 機能が必要です。



（注） これらの API は、サイト管理者とルート範囲の所有者のみが使用できます。

ロール オブジェクト

ロールオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	ロールの固有識別子。
app_scope_id	string	範囲が定義されている範囲。「サービスプロバイダーロール」の場合は空の場合があります。
name	string	ロールのユーザー指定名。
説明	string	ロールのユーザー指定の説明。

ロールの取得

このエンドポイントは、ユーザーがアクセス可能なロールのリストを返します。ロールは、特定のルート範囲にフィルタリングできます。範囲が指定されていない場合、ユーザーがアクセスできるすべての範囲のすべてのロールが返されます。サービス プロバイダ ロールは、ユーザーがサイト管理者である場合にのみ返されます。

`GET/openapi/v1/roles`

パラメータ：

名前	タイプ	説明
app_scope_id	string	(オプション) その範囲に割り当てられているロールのみを返すルート範囲の ID。

応答オブジェクト：ユーザーロールオブジェクトのリストを返します。

サンプル python コード

```
resp = restclient.get('/roles')
```

ロールの作成

このエンドポイントは、新しいロールを作成するために使用されます。

`POST/openapi/v1/roles`

パラメータ：

名前	タイプ	説明
name	string	ロールのユーザー指定名。
説明	string	ロールのユーザー指定の説明。
app_scope_id	string	(オプション) ロールに記載されている範囲 ID がサービス プロバイダ ロールと見なされない場合、ロールが作成される範囲 ID。

要求側のユーザーは、指定された範囲にアクセスできる必要があります。範囲のないロールは「サービス プロバイダ ロール」と呼ばれ、サイト管理者のみが作成できます。

応答オブジェクト：新しく作成されたロールオブジェクトを返します。

サンプル python コード

```
app_scope_id = '<app-scope-id>'
req_payload = {
      'name': 'Role Name',
      'description': 'Role Description',
      'app_scope_id': app_scope_id
}
restclient.post('/roles', json_body=json.dumps(req_payload))
```

特定のロールを取得

このエンドポイントは、特定のロール オブジェクトを返します。

```
GET/openapi/v1/roles/{role_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
role-id	string	ロールの固有識別子。

応答オブジェクト：指定された ID に関連付けられているロールオブジェクトを返します。

サンプル python コード

```
role_id = '<role-id>'
restclient.get('/roles/%s' % role_id)
```

ロールの更新

このエンドポイントは、既存のロールを更新するために使用されます。

```
PUT /openapi/v1/roles/{role_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
role-id	string	ロールの固有識別子。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
name	string	ロールのユーザー指定名。
説明	string	ロールのユーザー指定の説明。

要求側のユーザーは、指定された範囲にアクセスできる必要があります。範囲のないロールは「サービスプロバイダーロール」と呼ばれ、サイト管理者のみが更新できます。

応答オブジェクト：指定された ID を持つ更新されたロールオブジェクト。

サンプル python コード

```
role_id = '<role-id>'
req_payload = {
      'name': 'Role Name',
      'description': 'Role Description',
}
restclient.put('/roles/%s' % role_id, json_body=json.dumps(req_payload))
```

範囲へのロール アクセスを付与

このエンドポイントは、指定されたアクセス レベルを範囲に付与します。

```
POST/openapi/v1/roles/{role_id}/capabilities
```

機能は、ユーザーがアクセスできるロールにのみ追加できます。ロールが範囲に割り当てられている場合、機能はその範囲またはその子に対応している必要があります。サービス プロバイダ ロール (範囲に割り当てられていないもの) は、任意の範囲の機能を追加できます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
role-id	string	ロールの固有識別子。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
app_scope_id	string	アクセスが提供される範囲の ID。
能力	string	値は次のいずれかです。 SCOPE_READ, SCOPE_WRITE、EXECUTE、ENFORCE、 SCOPE_OWNER、DEVELOPER

機能の詳細については、「ロール」を参照してください。

応答オブジェクト：

名前	タイプ	説明
app_scope_id	string	アクセスが提供される範囲の ID。
role-id	string	ロールの ID。
能力	string	値は次のいずれかです。 SCOPE_READ、SCOPE_WRITE、EXECUTE、ENFORCE、SCOPE_OWNER、DEVELOPER
継承	boolean	

サンプル python コード

```
role_id = '<role-id>'
req_payload = {
  'app_scope_id': '<app-scope-id>',
  'ability': 'SCOPE_READ'
}
restclient.post('/roles/%s/capabilities' % role_id,
                json_body=json.dumps(req_payload))
```

特定のロールの削除

このエンドポイントは、指定された範囲を削除します。

```
DELETE /openapi/v1/roles/{role_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
role-id	string	ロールの固有識別子。

応答オブジェクト：なし。

サンプル python コード

```
role_id = '<role-id>'
restclient.delete('/roles/%s' % role_id)
```

ユーザ（Users）

この API のセットでユーザーを管理します。API キーに関連付けられた `user_role_scope_management` 機能が必要です。

（注） これらの API は、サイト管理者とルート範囲の所有者のみが使用できます。

ユーザー オブジェクト

ユーザー オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	ユーザー ロールの固有識別子。
電子メール	string	ユーザー アカウントに関連付けられている電子メール。
first_name	string	名前。

属性	タイプ	説明
last_name	string	姓。
app_scope_id	string	ユーザーに割り当てられる範囲。「サービス プロバイダー ユーザー」の場合、通常は空になります。
role_ids	リスト	ユーザー アカウントに割り当てられているロールの ID のリスト。
by-pass_external_auth	boolean	ローカルユーザーの場合は true、外部認証ユーザー（ldap または sso）の場合は false。
disabled_at	整数	ユーザーが無効になっているときの Unix タイムスタンプ。0 または null。そうでない場合。

ユーザーの取得

このエンドポイントは、Secure Workload アプライアンスに認識されているユーザーオブジェクトのリストを返します。

```
GET/openapi/v1/users
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
include_disabled	boolean	（オプション）無効なユーザーを含めるために、デフォルトで false に設定されます。
app_scope_id	string	（オプション）指定した範囲に割り当てられているユーザーのみを返します。

応答オブジェクト：ユーザーオブジェクトのリストを返します。サイト管理者のみが「サービス プロバイダ ユーザー」を表示できます。つまり、範囲に割り当てられていません。

サンプル python コード

```
GET /openapi/v1/users
```

新しいユーザ アカウントの作成

このエンドポイントは、新しいユーザー アカウントを作成するために使用されます。

`POST/openapi/v1/users`

パラメータ：JSON リクエストの本文には、次のパラメータが含まれます。

名前	タイプ	説明
電子メール	string	ユーザー アカウントに関連付けられている電子メール。
first_name	string	名前。
last_name	string	姓。
app_scope_id	string	(オプション) ユーザーが属するルート範囲。
role_ids	リスト	(オプション) ユーザーに割り当てられる必要があるロールのリスト。

app_scope_id は、ユーザーが割り当てられるルート範囲の ID です。app_scope_id が表示されない場合、そのユーザーは「サービス プロバイダー ユーザー」です。サイト管理者のみ、サービス プロバイダー ユーザーを作成できます。`role_ids` は、指定されたアプリケーション範囲の下で作成されたロールの ID です。

応答オブジェクト：新しく作成されたユーザーオブジェクトを返します。

サンプル python コード

```
req_payload = {
    "first_name": "fname",
    "last_name": "lname",
    "email": "foo@bar.com"
    "app_scope_id": "root_appscope_id",
    "role_ids": ["roleid1", "roleid2"]
}
resp = restclient.post('/users', json_body=json.dumps(req_payload))
```

特定のユーザーの取得

このエンドポイントは、特定のユーザー オブジェクトを返します。

`GET/openapi/v1/users/{user_id}`

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	ユーザーオブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられているユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'
resp = restclient.get('/users/%s' % user_id)
```

ユーザーの更新

このエンドポイントは、既存のユーザーを更新します。

```
PUT /openapi/v1/users/{user_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	更新されるユーザーオブジェクトの ID。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
電子メール	string	ユーザー アカウントに関連付けられている電子メール。
first_name	string	名前。
last_name	string	姓。
app_scope_id	string	ルートアプリケーション範囲ID（サイト管理者にのみ許可）

応答オブジェクト：新しく更新されたユーザーオブジェクトを返します。

サンプル python コード

```
req_payload = {
    "first_name": "fname",
    "last_name": "lname",
    "email": "foo@bar.com"
    "app_scope_id": "root_appscope_id",
}
restclient.put('/users', json_body=json.dumps(req_payload))
```

非アクティブ化されたユーザーの有効化/再アクティブ化

このエンドポイントは、非アクティブ化されたユーザーを再び有効化するために使用されます。

```
POST /openapi/v1/users/{user_id}/enable
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	有効化されているユーザーオブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられている、再アクティブ化されたユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'
resp = restclient.post('/users/%s/enable' % user_id)
```

ユーザー アカウントにロールを追加

このエンドポイントは、ユーザー アカウントにロールを追加するために使用されます。

`PUT/openapi/v1/users/{user_id}/add_role`

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	変更されるユーザーオブジェクトの ID。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
role-id	string	追加するロールオブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられている、変更されたユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'
req_payload = {
    "role_id": "5ce480d4497d4f1c155d0cef",
}
resp = restclient.put('/users/%s/add_role' % user_id,
                      json_body=json.dumps(req_payload))
```

ユーザー アカウントからロールを削除

このエンドポイントは、ユーザーアカウントからロールを削除するために使用されます。

`/Openapi/v1/users/{user_id}/remove_role の削除`

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	削除されるユーザーオブジェクトの ID。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
role-id	string	削除するロール オブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられている、変更されたユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'
req_payload = {
    "role_id": "5ce480d4497d4f1c155d0cef",
}
resp = restclient.delete('/users/%s/remove_role' % user_id,
                         json_body=json.dumps(req_payload))
```

指定されたユーザーの削除

このエンドポイントは、指定されたユーザー アカウントを削除します。

```
DELETE /openapi/v1/users/{user_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	削除されるユーザーオブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられている、削除されたユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'
resp = restclient.delete('/users/%s' % user_id)
```

インベントリ フィルタ

インベントリ フィルタは、インベントリ検索クエリの一致基準をエンコードします。この一連の API は、インベントリ フィルタ で説明されているものと同様の機能を提供します。API キーに関連付けられている `sensor_management or app_policy_management` 機能が必要です。

インベントリ フィルタ オブジェクト

インベントリ フィルタ JSON オブジェクトは、API エンドポイントに応じて、単一のオブジェクトまたはオブジェクトの配列として返されます。オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	インベントリフィルタの一意の識別子。
name	string	ユーザーが指定したインベントリフィルタの名前。
app_scope_id	string	フィルタに関連付けられた範囲の ID。
short_query{1}JSON{1}	JSON	フィルタに関連付けられているフィルタ (または一致基準)。
primary	boolean	「true」の場合は、フィルタが所有権の範囲に制限されていることを意味します。
public	boolean	「true」の場合、フィルタはその範囲にサービスを提供します。また、プライマリ/範囲が制限されている必要があります。
query	JSON	親範囲のフィルタと組み合わせて、フィルタに関連付けられているフィルタ (または一致基準)。[所有権の範囲に制限する（restricted to ownership scope）] チェックボックスがオンになっている場合に、これらの連携が有効になります。[primary (プライマリ)] フィールドが false の場合、クエリは short_query と同じです。

インベントリ フィルタの取得

このエンドポイントは、ユーザーに表示されるインベントリ フィルタのリストを返します。

GET/openapi/v1/filters/inventories

パラメータ：

名前	タイプ	説明
vrf-id	integer	VRF ID でインベントリフィルタを照合します。
root_app_scope_id	string	ルートアプリケーション範囲 ID でインベントリフィルタを照合します。
name	string	大文字と小文字を区別せずに、名前の一部に一致するインベントリフィルタを返します。
exact_name	string	大文字と小文字を区別して、正確な名前に一致するインベントリフィルタを返します。

インベントリ フィルタの作成

このエンドポイントは、インベントリ フィルタを作成するために使用されます。

POST/openapi/v1/filters/inventories

パラメータ：

名前	タイプ	説明
name	string	アプリケーション範囲のユーザー指定名。
query	JSON	フィルタに関連付けられているフィルタ(または一致基準)。
app_scope_id	string	フィルタに関連付けられた範囲の ID。
primary	boolean	「true」の場合は、フィルタが所有権の範囲に制限されていることを意味します。
public	boolean	「true」の場合、フィルタはその範囲にサービスを提供します。また、プライマリ/スコープが制限されている必要があります。

サンプル python コード

```
req_payload = {
  "app_scope_id": <app_scope_id>,
  "name": "sensor_config_inventory_filter",
  "query": {
    "type": "eq",
    "field": "ip",
    "value": <sensor_interface_ip>
  },
}
resp = restclient.post('/filters/inventories', json_body=json.dumps(req_payload))
```

インベントリフィルタクエリの検証

このエンドポイントは、必要なスキーマに対してクエリの構造を検証します。

```
POST /openapi/v1/filters/inventories/validate_query
```

パラメータ：

名前	タイプ	説明
query	JSON	範囲に関連付けられているフィルタ (または一致基準)。

応答オブジェクト：

属性	タイプ	説明
有効な	boolean	クエリが有効かどうかの表示
errors	アレイ	無効な場合、エラーの詳細

特定のインベントリ フィルタの取得

このエンドポイントは、インベントリフィルタのインスタンスを返します。

```
GET /openapi/v1/filters/inventories/{inventory_filter_id}
```

指定された ID に関連付けられているインベントリ フィルタ オブジェクトを返します。

特定のインベントリフィルタの更新

このエンドポイントは、インベントリフィルタを作成するために使用されます。

```
PUT /openapi/v1/filters/inventories/{inventory_filter_id}
```

パラメータ：

名前	タイプ	説明
name	string	ユーザーが指定した範囲の名前。
query	JSON	範囲に関連付けられているフィルタ (または一致基準)。
app_scope_id	string	フィルタに関連付けられた範囲の ID。
primary	boolean	「true」の場合は、フィルタが所有権の範囲に制限されていることを意味します。
public	boolean	「true」の場合、フィルタによって特定のサービスが提示されます。ポリシー生成の一部として使用できます。また、プライマリ/範囲が制限されている必要があります。

特定範囲の削除

このエンドポイントは、指定されたインベントリフィルタを削除します。

```
DELETE /openapi/v1/filters/inventories/{inventory_filter_id}
```

フロー検索

フロー検索機能は、「フロー」で説明されているのと同様の機能を提供します。これらの一連の API には、API キーに関連付けられている `flow_inventory_query` 機能が必要です。

フローの寸法のクエリ

このエンドポイントは、フロー検索クエリ (以下) に対して検索条件 (またはフィルタ) を指定できるフローカラムのリストを返します。列の詳細については、「列とフィルタ」を参照してください。

```
GET/openapi/v1/flowsearch/dimensions
```

パラメータ：（なし）

応答オブジェクト：

名前	タイプ	説明
寸法	文字列のリスト	アップロードされたユーザーとオーケストレータ ディメンションのリスト。

サンプル python コード

```
restclient.get('/flowsearch/dimensions')
```

フローメトリックのクエリ

このエンドポイントは、フロー観測に関連付けられたメトリックス (バイト数、パケット数など) のリストを返します。

`GET/openapi/v1/flowsearch/metrics`

パラメータ：（なし）

応答オブジェクト：

名前	タイプ	説明
メトリック	文字列のリスト	利用可能なメトリックのリスト

サンプル python コード

```
restclient.get('/flowsearch/metrics')
```

フローのクエリ

このエンドポイントは、フィルタ条件に一致するフローのリストを返します。結果の各フローオブジェクトには、フローの大きさ (上記のフロー寸法 API によって返される) とフロー メトリックス (上記のフロー メトリックス API によって返される) の結合である属性があります。

`POST/openapi/v1/flowsearch`

フィルタ条件で指定できるカラムのリストは、`/openapi/v1/flowsearch/ dimensions` API を使用して取得できます。

パラメータ：クエリ本文は、次のキーを使用した JSON 本文で構成されます。

名前	タイプ	説明
t0	整数または文字列	フロー検索開始時刻 (エポックまたは ISO 8601)
t1	整数または文字列	フロー検索終了時間 (エポックまたは ISO 8601)

名前	タイプ	説明
filter	JSON	クエリ フィルタ。フィルタが空(例: {}) の場合、クエリはすべてのフローに一致します。
scope-name	string	クエリが制限されている範囲の完全な名前。
寸法	array	(オプション) Flowsearch API の結果に返される寸法名のリスト。これは省略可能なパラメータです。指定しない場合、flowsearch の結果は使用可能なすべての寸法を返します。このオプションは、発信者が残りの寸法を気にしない場合に使用可能な寸法のサブセットを指定するときに役立ちます。
メトリック	array	(オプション) Flowsearch API の結果で返されるメトリックス名のリスト。これは省略可能なパラメータです。指定しない場合、flowsearch の結果は使用可能なすべてのメトリックスが返されます。このオプションは、発信者が他のメトリックスを処理しない場合に使用可能なメトリックスのサブセットを指定するのに役立ちます。
limit	integer	(オプション) 応答フローの制限数。
offset	string	(オプション) 前の応答から受信した Offset オブジェクト。
降順	boolean	(オプション) このパラメータが false または指定されていない場合、結果はタイムスタンプの昇順になります。パラメータ値が true の場合、結果はタイムスタンプの降順になります。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
    "t0": "2016-06-17T09:00:00-0700",
    "t1": "2016-06-17T17:00:00-0700",
    "filter": {
        "type": "and",
        "filters": [
            {
                "type": "contains",
                "field": "dst_hostname",
                "value": "prod"
            },
            {
                "type": "in",
                "field": "dst_port",
                "values": ["80", "443"]
            }
        ]
    },
    "scopeName": "Default:Production:Web",
    "limit": 100,
    "offset": <offset-object>
}
```

フィルタ

フィルタは、1つ以上のプリミティブフィルタで構成されるプリミティブフィルタと論理フィルタ(「not」、「and」、「or」)をサポートします。プリミティブフィルタの形式は次のとおりです。

```
{"type" : "<OPERATOR>", "field": "<COLUMN_NAME>", "value": "<COLUMN_VALUE>"}
```

プリミティブフィルタの場合は、`eq`、`ne`、`lt`、`lte`、`gt`や`gte`などの比較演算子を使用できます。演算子は、`in`、`regex`、`subnet`、`contains`や`range`にすることもできます。

プリミティブ フィルタの一部の例としては、次のものがあります。

```
{"type": "eq", "field": "src_address", "value": "7.7.7.7"}

{"type": "regex", "field": "src_hostname", "value": "prod.*"}

{"type": "subnet", "field": "src_addr", "value": "1.1.11.0/24"}

# Note, 'in' clause uses 'values' key instead of 'value'
{"type": "in", "field": "src_port", "values": [80, 443]}
```

not、and、orなどのブール演算を使用して、複雑なフィルタを指定することもできます。以下は、これらのタイプのフィルタの一部の例です。

```
# "and" and "or" operators need to specify list of "filters"
{"type": "and",
    "filters": [
        {"type": "in", "field": "src_port", "values": [80, 443]},
        {"type": "regex", "field": "src_hostname", "value": "prod.*"}
    ]
}
```

```
# "not" operator needs to specify a "filter"
{"type": "not",
    "filter": {"type": "subnet", "field": "src_addr", "value": "1.1.11.0/24"}
}
```

フロー検索要求での `filter` のスキーマは、より正式に示すと次のようになります。

キー	値
タイプ	フィルタ タイプ
field	プリミティブ フィルタのフィルタ フィールド カラム
filter	フィルタオブジェクト（フィルタタイプ `not` にのみ使用）
filters	フィルタオブジェクトのリスト（フィルタタイプ `and` および `or` に使用）
value	プリミティブ フィルタの値
値	`in` または `range` のフィルタタイプを使用したプリミティブフィルタの値のリスト

プリミティブ フィルタ タイプ

eq、neそれぞれ、「`field`」で指定されたカラム内の「`value`」で指定された値に等価または不等のフローを検索します。次のフィールドをサポートします。`src_hostname`、`dst_hostname`、`src_address`、`dst_address`、`src_port`、`dst_port`、`src_scope_name`、`dst_scope_name`、`vrf_name`、`src_enforcement_epg_name`、`dst_enforcement_epg_name`、`proto` これらの演算子は、ユーザーラベル付きカラムでも機能します。

lt、lte、gt、gte「`field`」で指定されたカラムの値が「`value`」で指定された値より小さい、値と等しい、値より大きい、値以上である（該当する場合）フローを検索します。次のフィールドをサポートします。`[src_port、dst_port]`

range「values」リストで指定された範囲の開始と範囲の終了の間で、「field」で指定されたカラムの値のフローを検索します（「values」リストは、「範囲」フィルタータイプのサイズ 2 である必要があります。最初の値は範囲の開始で、2番目の値は範囲の終了です）。次のフィールドをサポートします。[src_port、dst_port]

in 「`field`」で指定されたカラム内で、「`values`」で指定されたメンバーシップリストに一致するメンバーシップのフローを検索します。次のフィールドをサポートします。`src_hostname`、`dst_hostname`、`src_address`、`dst_address`、`src_port`、`dst_port`、`src_scope_name`、`dst_scope_name`、`vrf_name`、`src_enforcement_epg_name`、`dst_enforcement_epg_name`、`proto` この演算子は、ユーザーラベル付きカラムでも機能します。

regex、containsそれぞれ「`field`」で指定されたカラムで、「`value`」で指定された正規表現の正規表現一致または含む一致のフローを検索します。次のフィールドをサポートします。`src_hostname`、`dst_hostname`、`src_scope_name`、`dst_scope_name`、`vrf_name`、

src_enforcement_epg_name、dst_enforcement_epg_name これらの演算子は、ユーザーラベル付きカラムでも機能します。regex タイプのフィルタは、「value」として Java スタイルの正規表現パターンを使用する必要があります。

subnet CIDR 表記の文字列として「field」で指定されたサブネットメンバーシップのフローを検索します。次のフィールドをサポートしています。[src_address、dst_address]

論理フィルタ タイプ

not 「filter」で指定されたオブジェクトの論理 not フィルタ。

and 「filters」によって指定されたフィルタオブジェクトのリストの論理 and フィルタ

or 「filters」によって指定されたフィルタオブジェクトのリストの論理 or フィルタ

応答オブジェクト:

キー	値
offset	結果の次のページでパスされる応答オフセット
results	結果のリスト

結果の次のページを生成するには、offset の応答で受信したオブジェクトを取得し、それを次のクエリの offset の値として渡します。

サンプル python コード

```
   req_payload = {"t0": "2016-11-07T09:00:00-0700",
                  "t1": "2016-11-07T19:00:00-0700",
                  "scopeName": "Default:Prod:Web",
                  "limit": 10,
                  "filter": {"type": "and",
                      "filters": [
                          {"type": "subnet", "field": "src_address", "value":
"1.1.11.0/24"},
                          {"type": "regex", "field": "src_hostname", "value": "web*"}
                       ]
                  }
    }

    resp = restclient.post('/flowsearch', json_body=json.dumps(req_payload))
    print resp.status_code
    if resp.status_code == 200:
        parsed_resp = json.loads(resp.content)
        print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

フローの TopN クエリ

このエンドポイントは、リスト内のランクが指定されたメトリックスの集約によって決定される場合に、指定された寸法値の上位 n 個のソートされたリストを返します。

```
POST/openapi/v1/flowsearch/topn
```

パラメータ：

フィルタ条件で指定できるカラムのリストは、`/openapi/v1/flowsearch/ dimensions` API を使用して取得できます。要求の本文は、JSON 形式のクエリである必要があります。次に、クエリ本文の例を示します。要求本文内のパラメータ `t0` および `t1` は、エポック形式または iso8601 形式で指定できます。TopN API では、1 日の最大時間範囲のクエリのみが可能です。グループ化を実行する必要があるディメンションは、 `dimension` を使用して指定する必要があります。上位 N 件の結果をランク付けする必要があるメトリックスは、JSON 本文の `metric` フィールドで指定する必要があります。`threshold` には、「上位 N 件」の「N」を示す 1 以上の値を 指定する必要があります。`threshold` の最大値は 1000 です。ユーザーが 1000 を超える値を指定した場合でも、API は最大で 1000 件の結果のみを返します。また、`scopeName` パラメータを指定する必要があります。これは、検索対象を絞り込むための完全な範囲名です。`filter` は、フロー検索（「フィルタ （1147ページ）」）のフィルタと同じです。`filter` を指定しない場合、上位 N 件はすべてのフローエントリが対象になります。

```
    {
      "t0": "2016-06-17T09:00:00-0700",    # t0 can also be 1466179200
      "t1": "2016-06-17T17:00:00-0700",    # t1 can also be 1466208000
      "dimension": "src_address",
      "metric": "fwd_pkts",
      "filter": {"type": "eq", "field": "src_address", "value": "172.29.203.193"},
#optional
      "threshold": 5,
      "scopeName": "Default"
    }
```

クエリ本文は、次のキーを使用した JSON 本文で構成されます。

キー	値
t0	フローの開始時刻 (エポックまたは ISO 8601)
t1	フローの終了時刻 (エポックまたは ISO 8601)
filter	クエリ フィルタ。フィルタが空 (例: {})、またはフィルタが存在しない (オプション) 場合、topN クエリはすべてのフローエントリに適用されます。
scope-name	範囲の完全な名前。クエリの実行対象が指定した範囲のみになります。
dimension	dimension は、グループ化で使用するフィールドです。
metric	metric は dimension の値の合計数です。
threshold	threshold は上位 N 件の「N」の部分です。

応答オブジェクト:

キー	値
result	上位 N エントリの配列

サンプル python コード

```
req_payload = {
    "t0": "2017-06-07T08:20:00-07:00",
    "t1": "2017-06-07T14:20:00-07:00",
    "dimension": "src_address",
    "metric": "fwd_pkts",
    "filter": {"type": "ne", "field": "src_address", "value": "172.29.203.193"},
    "threshold": 5,
    "scopeName": "Default"
}
resp = rc.post('/flowsearch/topn',
               json_body=json.dumps(req_payload))
print resp.status_code
if resp.status_code == 200:
   parsed_resp = json.loads(resp.content)
   print json.dumps(parsed_resp)
```

サンプル応答

```
[
  { "result": [
    {"src_address": "172.31.239.163", "fwd_pkts": 23104},
    {"src_address": "172.31.239.162", "fwd_pkts": 22410},
    {"src_address": "172.31.239.166", "fwd_pkts": 16185},
    {"src_address": "172.31.239.168", "fwd_pkts": 15197},
    {"src_address": "172.31.239.169", "fwd_pkts": 15116}
    ]
  }
]
```

フローカウント

このエンドポイントは、指定された基準に一致するフロー観測の数を返します。

`POST/openapi/v1/flowsearch/count`

パラメータ：

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。要求本文内のパラメータ `t0` および `t1` は、エポック形式または iso8601 形式で指定できます。この API は、1 日の最大時間範囲のクエリのみを許可します。また、`scopeName` パラメータを指定する必要があります。これは、検索を制限する範囲の完全な名前です。このパラメータが指定されていない場合、フロー観測カウント API 要求は、読み取りアクセス権を持つすべての範囲に適用されます。`filter` は、フロー検索フィルタのフィルタのものと同じです。

```
  {
    "t0": "2016-06-17T09:00:00-0700",    # t0 can also be 1466179200
    "t1": "2016-06-17T17:00:00-0700",    # t1 can also be 1466208000
    "filter": {"type": "eq", "field": "src_address", "value": "172.29.203.193"},
```

```
  "scopeName": "Default"
}
```

クエリ本文は、次のキーを使用した JSON 本文で構成されます。

キー	値
t0	フローの開始時刻 (エポックまたは ISO 8601)
t1	フローの終了時刻 (エポックまたは ISO 8601)
filter	クエリ フィルタ。フィルタが空 (例: {}) の場合、クエリはすべてのフローに一致します。
scope-name	範囲の完全な名前。クエリの実行対象が指定した範囲のみになります。

応答オブジェクト:

キー	値
count	フロー検索条件に一致するフロー観測の数。

サンプル python コード

```
req_payload = {
      "t0": "2017-07-20T08:20:00-07:00",
      "t1": "2017-07-20T10:20:00-07:00",
      "scopeName": "Tetration",
      "filter": {
                  "type": "eq",
                  "field": "dst_port",
                  "value": "5642"
          }
}
resp = rc.post('/flowsearch/count',
               json_body=json.dumps(req_payload))
print resp.status_code
if resp.status_code == 200:
   parsed_resp = json.loads(resp.content)
   print json.dumps(parsed_resp)
```

サンプル応答

```
{"count":508767}
```

インベントリ

インベントリ検索APIは、インベントリ検索で説明されているのと同様の機能を提供します。これらの一連のAPIには、APIキーに関連付けられている `flow_inventory_query` 機能が必要です。

インベントリの寸法のクエリ

このエンドポイントは、インベントリ検索クエリに対して検索条件(またはフィルタ)を指定できるインベントリ カラムのリストを返します。

```
GET/openapi/v1/inventory/search/dimensions
```

インベントリ検索

このエンドポイントは、指定された条件に一致するインベントリ項目のリストを返します。

```
POST/openapi/v1/inventory/search
```

フィルタ条件で指定できるカラムのリストは、/openapi/v1/inventory/ search/dimensions API を使用して取得できます。

パラメータ：

名前	タイプ	説明
filter	JSON	フィルタ クエリ。
scope-name	string	(オプション) 結果を制限する範囲の名前。
limit	integer	（オプション）返される結果の最大数。
offset	integer	(オプション) 次のページを取得する以前の要求からのオフセット。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
    "filter": {
        "type": "contains",
        "field": "hostname",
        "value": "collector"
    },
    "scopeName": "Default:Production:Web",  // optional
    "limit": 100,
    "offset": "<offset-object>"               // optional
}
```

サポートされているさまざまなタイプのフィルタを取得するには、フィルタ （1147ページ） を参照してください。

クエリ本文は、次のキーを使用した JSON 本文で構成されます。

キー	値
filter	クエリ フィルタ。フィルタが空 (例: {}) の場合、クエリはすべてのインベントリ項目に一致します。
scope-name	クエリが制限される範囲の完全な名前 (オプション)
寸法	インベントリ検索 API の結果で返される寸法名のリスト。これは省略可能なパラメータです。指定しない場合、結果として使用可能なすべての寸法が返されます。このオプションは、発信者が残りの寸法を気にしない場合に使用可能な寸法のサブセットを指定するときに役立ちます。
制限	応答項目の制限数（オプション）
offset	前の応答から受信したオフセットオブジェクト（オプション）

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

名前	タイプ	説明
offset	integer	結果の次のページに渡される応答オフセット。
results	オブジェクトの配列	結果のリスト。

応答には、ページ分割された応答の `offset` フィールドが含まれる場合があります。ユーザーは、次の一連の結果を取得するために、後続の要求で同じオフセットを指定する必要があります。

サンプル Python コード

```
req_payload = {
  "scopeName": "Tetration", # optional
  "limit": 2,
  "filter": {"type": "and",
     "filters": [
        {"type": "eq", "field": "vrf_name", "value": "Tetration"},
        {"type": "subnet", "field": "ip", "value": "1.1.1.0/24"},
        {"type": "contains", "field": "hostname", "value": "collector"}
     ]
  }
}

resp = restclient.post('/inventory/search', json_body=json.dumps(req_payload))
print resp.status_code
```

```
if resp.status_code == 200:
   parsed_resp = json.loads(resp.content)
   print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

インベントリ統計情報

このエンドポイントは、インベントリ項目の統計情報を返します。

```
GET /openapi/v1/inventory/{id}/stats?t0=<t0>&t1=<t1>&td=<td>
```

表 *57:*

パス パラメータ	説明
id	Inventory item id as `{ip}-{vrf_id}` such as `1.1.1.1-123`

クエリー パラメータ	説明
t0	エポック時間の統計情報の開始時刻
t1	エポック時間の統計情報の終了時刻
td	統計情報集約の粒度。整数で秒数を指定します。「分」、「時間」、「日」などの文字列をパスできます。

サンプル python コード

```
resp = restclient.get('/inventory/1.1.1.1-123/stats?t0=1483228800&t1=1485907200&td=day')
```

インベントリ カウント

このエンドポイントは、指定された条件に一致するインベントリ項目のリストを返します。

```
POST/openapi/v1/inventory/count
```

フィルタ条件で指定できるカラムのリストは、`/openapi/v1/inventory/ search/dimensions` API を使用して取得できます。

パラメータ：

名前	タイプ	説明
filter	JSON	フィルタ クエリ。
scope-name	string	(オプション) 結果を制限する範囲の名前。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
    "filter": {
        "type": "and",
        "filters": [
            {
                "type": "contains",
                "field": "hostname",
                "value": "prod"
            },
            {
                "type": "subnet",
                "field": "ip"
                "value": "6.6.6.0/24"
            }
        ]
    },
    "scopeName": "Default:Production:Web",  # optional
}
```

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

表 *58 :*

キー	値
count	フィルタ条件に一致するインベントリ項目の数

サンプル python コード

```
req_payload = {
  "scopeName": "Tetration", # optional
  "filter": {"type": "and",
     "filters": [
        {"type": "eq", "field": "vrf_name", "value": "Tetration"},
        {"type": "subnet", "field": "ip", "value": "1.1.1.0/24"},
        {"type": "contains", "field": "hostname", "value": "collector"}
     ]
  }
}

resp = restclient.post('/inventory/count', json_body=json.dumps(req_payload))
print resp.status_code
if resp.status_code == 200:
   parsed_resp = json.loads(resp.content)
   print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

インベントリの脆弱性

このエンドポイントは、脆弱なワークロードに関連付けられた IP アドレスに対応する CVE を返します。

この API は、ルート範囲への最小限の読み取りアクセス権を持つユーザーのみが使用できます。

```
POST /openapi/v1/inventory/cves/{rootScopeID}
```

パラメータ：

名前	タイプ	説明
ips	文字列のリスト	CVE情報を取得するIPのリスト。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
    {
        "ips": [
            "10.18.187.72",
            "10.18.187.73"
        ]
    }
```

応答

この応答は、本文の JSON オブジェクトの配列で、次のプロパティが含まれます。

名前	タイプ	説明
ip	string	IP アドレス
cve_ids	文字列のリスト	IP アドレスを持つインベントリの CVE ID のリスト。

サンプル python コード

```
  root_scope_id = "5fa0d242497d4f7d968c669b"
  req_payload = {
    "ips":["10.18.187.72", "10.18.187.73"]
  }

  resp = restclient.post('/inventory/cves/' + root_scope_id,
json_body=json.dumps(req_payload))
  print resp.status_code
  if resp.status_code == 200:
     parsed_resp = json.loads(resp.content)
     print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

ワークロード

ワークロード API は、「ワークロード プロファイル」ページのコンテンツへの、プログラムによるアクセスを提供します。これらの一連の API には、API キーに関連付けられている `sensor_management or flow_inventory_query` 機能が必要です。

ワークロードの詳細

このエンドポイントは、特定のワークロードの指定されたエージェント UUID を返します。

```
GET /openapi/v1/workload/{uuid}
```

パス パラメータ	説明
uuid	エージェント UUID

応答

応答は、指定されたUUIDに関連付けられたワークロードオブジェクトです。ワークロードオブジェクトの属性スキーマを以下に示します。

表 *59 :*

属性	タイプ	説明
agent_type	string	エージェント タイプ
auto_upgrade_opt_out	boolean	trueの場合、エージェントはクラスタのアップグレード時に自動的にアップグレードされません。
cpu_quota_mode	integer	CPU クォータ制御
cpu_quota_us	integer	CPU クォータ使用率
current_sw_version	string	ワークロードで実行されているエージェントソフトウェアのバージョン
data_plane_disabled	boolean	trueの場合、フローテレメトリデータはエージェントからクラスタにエクスポートされません。
desired_sw_version	string	ワークロードでの実行を意図したエージェントソフトウェアのバージョン
enable_conversation_mode	boolean	trueの場合、カンバセーションモードが有効になります。
enable_cache_sidechannel	boolean	trueの場合、サイドチャネル攻撃の検出が有効になります。
enable_forensics	boolean	trueの場合、フォレンジックが有効になります。

属性	タイプ	説明
enable_meltdown	boolean	trueの場合、メルトダウンエクスプロイト検出が有効になります。
enable_pid_lookup	boolean	true の場合、プロセスルックアップが有効になります。
forensics_cpu_quota_mode	integer	フォレンジック CPU クォータ制御
forensics_cpu_quota_us	integer	フォレンジッククォータの使用状況
forensics_mem_quota_bytes	integer	フォレンジック メモリ クォータ（バイト単位）
host_name	string	ワークロードのホスト名
interfaces	アレイ	インターフェイス オブジェクトの配列
kernel_version	string	カーネル バージョン（Kernel version）
last_config_fetch_at	integer	最終設定がフェッチされた日時
last_software_update_at	integer	最終ソフトウェアは、エージェントが現在のバージョンを報告したときのタイムスタンプです。
max_rss_limit	integer	最大メモリ制限
platform	string	ワークロードのプラットフォーム
uuid	string	エージェントの一意の ID
windows_enforcement_mode	string	Windows 強制モードの種類。WAF（Windows Advanced Firewall）または WFP（Windows Filtering Platform）

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'
resp = restclient.get('/workload/%s' % (agent_uuid))
```

ワークロードの統計情報

このエンドポイントは、ワークロードの統計情報を返します。

```
GET /openapi/v1/workload/{uuid}/stats?t0=<t0>&t1=<t1>&td=<td>
```

パス パラメータ	説明
uuid	エージェント UUID

クエリ URL には次のパラメータが含まれています

クエリー パラメータ	説明
t0	エポック時間の統計情報の開始時刻
t1	エポック時間の統計情報の終了時刻終了時刻は開始時刻を 1 日以上超えることができません。
td	統計情報集約の粒度。整数で秒数を指定します。「分」、「時間」、「日」などの文字列をパスできます。

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

名前	タイプ	説明
timestamp	string	メトリックが収集された時刻（エポックまたは ISO 8601）
results	オブジェクト	メトリック

メトリックは、次のプロパティを持つ JSON オブジェクトです

名前	タイプ	説明
flow_count	integer	フロー数。
rx_byte_count	integer	受信バイト数。
rx_packet_count	integer	受信パケット数。
tx_byte_count	integer	送信バイト数。
tx_packet_count	integer	送信パケット数。

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'
```

```
  td = 15 * 60 # 15 minutes
  resp = restclient.get('/workload/%s/stats?t0=1483228800&t1=1485907200&td=%d' %
(agent_uuid, td))

  # This code queries workload statistics for a week
  t0 = 1483228800
  for _ in range(7):
    t1 = t0 + 24 * 60 * 60
    resp = restclient.get('/workload/%s/stats?t0=%d&t1=%d&td=day' % (agent_uuid, t0,
t1))
    t0 = t1
```

インストールされたソフトウェアパッケージ

このエンドポイントは、ワークロードにインストールされているパッケージのリストを返します。

```
GET /openapi/v1/workload/{uuid}/packages
```

パス パラメータ	説明
uuid	エージェント UUID

応答

Thw 応答は、パッケージ JSON オブジェクトの配列です。パッケージオブジェクトのスキーマは次のとおりです。

属性	タイプ	説明
アーキテクチャ	string	パッケージのアーキテクチャ
name	string	パッケージの名前
パブリッシャ	string	パッケージの発行元
version	string	パッケージのバージョン

サンプル python コード

```
  agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'
  resp = restclient.get('/workload/%s/packages' % (agent_uuid))
```

ワークロードの脆弱性

このエンドポイントは、ワークロードで観察された脆弱性のリストを返します。

```
GET /openapi/v1/workload/{uuid}/vulnerabilities
```

脆弱性オブジェクトは、次のキーを使用した JSON 本文で構成されます。

パス パラメータ	説明
uuid	エージェント UUID

応答

応答は、脆弱性 JSON オブジェクトの配列です。脆弱性オブジェクトのスキーマは次のとおりです。

属性	タイプ	説明
cve_id	string	Common Vulnerability Exposure の ID
package_infos	アレイ	パッケージ情報オブジェクトの配列
v2_score	float	CVSS 2 スコア
v2_access_complexity	string	CVSS V2 アクセスの複雑さ
v2_access_vector	string	CVSS V2 アクセスベクトル
v2_authentication	string	CVSS V2 認証
v2_availability_impact	string	CVSS V2 可用性への影響
v2_confidentiality_impact	string	CVSS V2 機密性への影響
v2_integrity_impact	string	CVSS V2 完全性への影響
v2_severity	string	CVSS V2 重大度
v3_score	float	CVSS V3 スコア
v3_attack_complexity	string	CVSS V3 攻撃の複雑さ
v3_attack_vector	string	CVSS V3 攻撃ベクトル
v3_availability_impact	string	CVSS V3 可用性への影響
v3_base_severity	string	CVSS V3 基本重大度
v3_confidentiality_impact	string	CVSS V2 機密性への影響
v3_integrity_impact	string	CVSS V3 完全性への影響
v3_privileges_required	string	必要な CVSS V3 権限
v3_scope	string	CVSS V3 範囲
v3_user_interaction	string	CVSS V3 ユーザーの操作

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'
resp = restclient.get('/workload/%s/vulnerabilities' % (agent_uuid))
```

ワークロードの長時間実行プロセス

このエンドポイントは、ワークロードで長時間実行されているプロセスのリストを返します。長時間実行プロセスは、稼働時間が 5 分以上のプロセスとして定義されます。

```
GET /openapi/v1/workload/{uuid}/process/list
```

パス パラメータ	説明
uuid	エージェント UUID

応答

応答は、プロセス JSON オブジェクトのリストです。

属性	タイプ	説明
cmd	string	プロセスのコマンド文字列
binary_hash	string	16 進数のプロセスバイナリの SHA256
ctime	long	使用中のプロセスバイナリの ctime
mtime	long	使用中のプロセスバイナリの mtime
exec_path	string	プロセスの実行パス
exit_usec	long	プロセスが使用中に終了した時刻
num_libs	integer	プロセスでロードされるライブラリの数
pid	integer	[プロセス ID（Process ID）]
ppid	integer	親プロセス ID
pkg_info_name	string	プロセスに関連付けられたパッケージの名前
pkg_info_version	string	プロセスに関連付けられたパッケージのバージョン
proc_state	string	プロセスの状態
uptime	long	使用中のプロセスの稼働時間
username	string	プロセスのユーザー名

属性	タイプ	説明
resource_usage	アレイ	リソース使用状況の配列 オブジェクト

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'
resp = restclient.get('/openapi/v1/workload/%s/process/list' % (agent_uuid))
```

ワークロード プロセス スナップショットのサマリー

このエンドポイントは、ワークロード プロセス スナップショットのサマリーを返します。プロセススナップショットには、特定の時間にワークロードによってキャプチャされたすべてのプロセスが含まれています。現在、最新のプロセススナップショットのコピーが 1 つ保持されています。エンドポイントは、将来の拡張を容易にするために、空のペイロードを持つ POST メソッドをサポートします。

```
POST /openapi/v1/workload/{uuid}/process/tree/ids
```

パス パラメータ	説明
uuid	エージェント UUID

応答

応答は、プロセススナップショットのサマリー JSON オブジェクトのリストです。

属性	タイプ	説明
sensor_uuid	string	エージェント UUID
handle	string	取得するプロセススナップショットへのハンドル
process_count	integer	スナップショットのプロセス数
ts_usec	integer	スナップショットがキャプチャされたときのタイムスタンプ

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'
payload = {
}
resp = restclient.post('/openapi/v1/workload/%s/process/tree/ids' %
                       agent_uuid, json_body=json.dumps(payload))
```

ワークロード プロセス スナップショット

このエンドポイントは、ワークロードのプロセススナップショットを返します。プロセススナップショットには、特定の時間にワークロードによってキャプチャされたすべてのプロセスが含まれています。現在、最新のプロセススナップショットのコピーが 1 つ保持されています。このエンドポイントは、ワークロード プロセス スナップショットのサマリーエンドポイントと一緒に使用する必要があります。

```
POST /openapi/v1/workload/{uuid}/process/tree/details
```

パス パラメータ	説明
uuid	エージェント UUID

ペイロードフィールド	タイプ	説明
handle	string	取得するプロセススナップショットへのハンドル

応答

応答は、JSON でのスナップショットに属するプロセスのリストです。

属性	タイプ	説明
command-string	string	トークン化されたコマンド文字列
command_string_raw	string	raw コマンド文字列
binary_hash	string	16 進数のプロセスバイナリの SHA256
ctime	long	使用中のプロセスバイナリの ctime
mtime	long	使用中のプロセスバイナリの mtime
exec_path	string	プロセスの実行パス
process_id	integer	[プロセス ID（Process ID）]
parent_process_id	integer	親プロセス ID
process_key	integer	プロセスの一意のキー
parent_process_key	integer	親プロセスへの一意のキー
pkg_info_name	string	プロセスに関連付けられたパッケージの名前

属性	タイプ	説明
pkg_info_version	string	プロセスに関連付けられたパッケージのバージョン
proc_state	string	プロセスの状態
uptime	long	使用中のプロセスの稼働時間
username	string	プロセスのユーザー名
cve_ids	アレイ	CVEID オブジェクトの配列

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'
payload = {
}
resp = restclient.post('/openapi/v1/workload/%s/process/tree/ids' %
            agent_uuid, json_body=json.dumps(payload))
handle = json.loads(resp.text)['process_summary'][0]['summary'][0]['handle']
payload = {
  "handle": handle,
}
resp = restclient.post('/openapi/v1/workload/%s/process/tree/details' %
            agent_uuid, json_body=json.dumps(payload))
```

JSON オブジェクトの定義：

インターフェイス

属性	タイプ	説明
ip	string	インターフェイスの IP アドレス
mac	string	インターフェイスの MAC アドレス
name	string	インターフェイスの名前
netmask	string	インターフェイスのネットマスク
pcap_opened	boolean	false の場合、インターフェイスのパケットキャプチャは有効化されていません
tags_scope_id	アレイ	インターフェイスに関連付けられた範囲 ID

属性	タイプ	説明
vrf	string	[VRF名（VRF Name）]
vrf-id	整数	VRF ID

パッケージ情報

属性	タイプ	説明
name	string	パッケージ名
version	string	パッケージのバージョン

リソース使用状況

属性	タイプ	説明
cpu_usage	float	CPU 使用率
memory_usage_kb	integer	メモリ使用量
ts_usec	long	リソース使用量がキャプチャされたときのタイムスタンプ

CVE ID

属性	タイプ	説明
cve_id	string	cve ID
impact__cvss_v2__access_complexity	string	CVE アクセスの複雑度
impact__cvss_v2__access_vector	string	CVE access vector

デフォルトのポリシー生成設定

この一連の API は、ルート範囲のデフォルトのポリシー生成設定を読み取り、更新するために使用されます。

この API には、API キーに関連付けられている `app_policy_management` 機能が必要です。

（注） これらの API は、サイト管理者とルート範囲の所有者のみが使用できます。

- ポリシー生成設定オブジェクト （1168 ページ）

• デフォルトのポリシー生成設定の取得 （1169 ページ）
• デフォルトのポリシー生成設定の設定 （1170 ページ）

ポリシー生成設定オブジェクト

属性	タイプ	説明
carry_over_policies	boolean	承認済みとしてマークされたポリシーは、可能な場合は維持されます
deep_policy_generation	boolean	指定された範囲の下に範囲ツリー全体のポリシーを作成します（指定された範囲内のすべてのメンバーを含む）
skip_clustering	boolean	クラスタリングをスキップするには true に設定します。既存の承認済みクラスタと範囲でポリシーが生成されます
auto_accept_policy_connectors	boolean	すべての発信ポリシーコネクタを自動承諾します
enable_exclusion_filter	boolean	入力フローデータに除外フィルタを適用します
enable_default_exclusion_filter	boolean	入力フローデータにデフォルトの除外フィルタを適用します
remove_redundant_policies	boolean	ディープポリシー生成中に冗長ポリシーを削除します
enable_service_discovery	boolean	false に設定すると、センサーによって報告された adm パイプラインのエフェメラルポート範囲に基づいて、ポリシー生成がスキップされます。現在は、Windows Active Directory のポリシーの生成に使用されています。
externals	array	外部依存関係オブジェクトの順序付きリスト
clustering_granularity	string	VERY_COARSE、COARSE、MEDIUM、FINE、VERY_FINE のいずれか

属性	タイプ	説明
policy_compression	string	DISABLED、CONSERVATIVE、MODERATE、AGGRESSIVE、VERY_AGGRESSIVE のいずれか
port_generalization	string	DISABLED、CONSERVATIVE、MODERATE、AGGRESSIVE、VERY_AGGRESSIVE のいずれか
sim_policy	integer	1 => フロー、2 => プロセス、5 => 両方

外部依存関係オブジェクト

名前	タイプ	説明
id	string	フィルタの ID
filter_type	string	AppScope または UserInventoryFilter
incluce	array	有効にする user_filters boolean と、提供されたサービス インベントリ フィルタの順序付きリストの user_filter_list を持つオブジェクト

•

デフォルトのポリシー生成設定の取得

このエンドポイントは、現在のデフォルトのポリシー生成設定を返します。何も作成されていない場合は、空のオブジェクトを返すことができます。

```
GET /openapi/v1/app_scopes/default_adm_run_config
```

パラメータ：

リクエスト URL には次のパラメータが含まれています

名前	タイプ	説明
root_app_scope_id	string	このデフォルト設定が適用されるルート範囲の固有識別子

応答オブジェクト：現在のデフォルトのポリシー生成設定を返します。何も作成されていない場合は、空のオブジェクトを返します。

デフォルトのポリシー生成設定の設定

このエンドポイントは、デフォルトのポリシー生成設定を設定します。

```
PUT /openapi/v1/app_scopes/default_adm_run_config
```

上記のポリシー生成設定オブジェクトリストの値に加えて、次のパラメータがあります。

名前	タイプ	説明
root_app_scope_id	string	このデフォルト設定が適用されるルート範囲の固有識別子

応答オブジェクト：デフォルトのポリシー生成設定を返します。応答オブジェクト：デフォルトのポリシー生成設定を返します。

フォレンジックインテント

このソフトウェアエージェントAPIは、フォレンジックインテントの管理に関連付けられています。

フォレンジックインテントは、フォレンジックプロファイルを適用先のエージェントのグループにリンクします。エージェントのグループは、インベントリフィルタを使用して定義します。

これらのAPIのセットには、APIキーに関連付けられている `sensor_management` 機能が必要です。

（注） これらのAPIは、サイト管理者とルート範囲の所有者のみが使用できます。

フォレンジック インテント オブジェクト

属性	タイプ	説明
id	string	インテントの固有識別子
name	string	インテントの名前
inventory_filter_id	array	このインテントに関連付けられているインベントリフィルタのID
forensic_config_profile_id	integer	このインテントに関連付けられているプロファイルのID

属性	タイプ	説明
created_At	整数	インテントが作成されたときの Unix タイムスタンプ
updated_at	整数	インテントが最後に更新されたときの Unix タイムスタンプ

フォレンジックインテントのリスト

このエンドポイントは、既存のすべてのフォレンジックプロファイルをリストします。

```
GET /openapi/v1/inventory_config/forensic_intents
```

パラメータ：（なし）

このエンドポイントは、フォレンジック インテント オブジェクトの概要の配列を返します。

単一のフォレンジックインテントの取得

```
GET /openapi/v1/inventory_config/forensic_intents/{intent_id}
```

パラメータ：

名前	タイプ	説明
intent_id	string	インテントの ID

フォレンジック インテント オブジェクトの詳細を返します。

フォレンジックインテントの作成

```
POST /openapi/v1/inventory_config/forensic_intents
```

パラメータ：

名前	タイプ	説明
name	string	インテントの名前
inventory_filter_id	string	このインテントに関連付けられているインベントリフィルタの ID
forensic_config_profile_id	string	このインテントに関連付けられているフォレンジックプロファイルの ID

フォレンジック インテント オブジェクトを返します。

フォレンジックインテントの更新

```
PUT /openapi/v1/inventory_config/forensic_intents/{intent_id}
```

パラメータ：

名前	タイプ	説明
intent_id	string	インテントの ID
name	string	インテントの名前
inventory_filter_id	string	このインテントに関連付けられているインベントリフィルタの ID
forensic_config_profile_id	string	このインテントに関連付けられているフォレンジックプロファイルの ID

フォレンジック インテント オブジェクトを返します。

フォレンジックインテントの削除

```
DELETE /openapi/v1/inventory_config/forensic_intents/{intent_id}
```

パラメータ：

名前	タイプ	説明
intent_id	string	インテントの ID

成功すると 200 を返します。

フォレンジックインテント順序

このソフトウェアエージェントAPIは、フォレンジックインテント順序の管理に関連付けられています。

フォレンジックプロファイルは、インテントを使用してエージェントに適用されます。インテントは、インベントリフィルタを使用してエージェントのグループを定義します。フィルタが重複している場合は、どれが適用されるかを把握する必要があります。インテントの優先順位を定義するために順序を使用します。

これらの API のセットには、API キーに関連付けられている `'sensor_management'` 機能が必要です。

（注） これらの API は、サイト管理者とルート範囲の所有者のみが使用できます。

- フォレンジックインテント順序オブジェクト （1173 ページ）
- 現在のフォレンジックインテント順序の取得 （1173 ページ）
- フォレンジックインテントの作成 （1171 ページ）

フォレンジックインテント順序オブジェクト

属性	タイプ	説明
version	string	現在の順序のバージョン
intents	array	順序内のインテントオブジェクトの名前
intent_ids	array	フォレンジックインテント ID の配列

現在のフォレンジックインテント順序の取得

このエンドポイントは、現在のフォレンジックインテント順序を返します。

```
GET /openapi/v1/inventory_config/forensic_orders
```

パラメータ：（なし）

このエンドポイントは、現在のフォレンジックインテント順序オブジェクトを返します。

フォレンジックインテント順序の作成

```
POST /openapi/v1/inventory_config/forensic_orders
```

パラメータ：

名前	タイプ	説明
version	string	現在の順序と一致する必要があります
intent_ids	array	インテント ID の配列

フォレンジックインテント順序オブジェクトを返します。

フォレンジックプロファイル

このソフトウェアエージェントAPIは、フォレンジックプロファイルの管理に関連付けられています。

フォレンジックプロファイルは、フォレンジックインテントを使用してエージェントのグループに適用できる、ルールのコレクションです。

これらの API のセットには、API キーに関連付けられている `'sensor_management'` 機能が必要です。



（注） これらの API は、サイト管理者とルート範囲の所有者のみが使用できます。

- フォレンジック プロファイル オブジェクト （1174 ページ）
- フォレンジックプロファイルのリスト （1175 ページ）
- 単一のフォレンジックプロファイルの取得 （1175 ページ）
- フォレンジックプロファイルの作成 （1175 ページ）
- フォレンジックプロファイルの更新 （1175 ページ）
- フォレンジックプロファイルの削除 （1176 ページ）

フォレンジック プロファイル オブジェクト

属性	タイプ	説明
id	string	プロファイルの固有識別子
name	string	プロファイルの名前
forensic_rules	array	プロファイルに関連付けられたルールの配列
created_At	integer	プロファイルが作成されたときの Unix タイムスタンプ
updated_at	integer	プロファイルが最後に更新されたときの Unix タイムスタンプ
is_readonly	boolean	プロファイルが読み取り専用かどうかを示します

属性	タイプ	説明
root_app_scope_id	string	プロファイルが属するルート範囲の ID

フォレンジックプロファイルのリスト

このエンドポイントは、既存のすべてのフォレンジックプロファイルをリストします。

```
GET /openapi/v1/inventory_config/forensic_profiles
```

パラメータ：（なし）

このエンドポイントは、フォレンジック プロファイル オブジェクトの概要の配列を返します。

単一のフォレンジックプロファイルの取得

```
GET /openapi/v1/inventory_config/forensic_profiles/{profile_id}
```

パラメータ：

名前	タイプ	説明
profile_id	string	プロファイルの ID

フォレンジック プロファイル オブジェクトの詳細を返します。

フォレンジックプロファイルの作成

```
POST /openapi/v1/inventory_config/forensic_profiles
```

パラメータ：

名前	タイプ	説明
name	string	プロファイルの名前
root_app_scope_id	string	プロファイルが属するルート範囲の ID
forensic_rule_ids	array	このプロファイルに関連付けるフォレンジックルール ID の配列

フォレンジック プロファイル オブジェクトを返します。

フォレンジックプロファイルの更新

```
PUT /openapi/v1/inventory_config/forensic_profiles/{id}
```

パラメータ：

名前	タイプ	説明
profile_id	string	プロファイルの ID
name	string	プロファイルの名前
forensic_rule_ids	array	このプロファイルに関連付けるフォレンジックルール ID の配列

フォレンジック プロファイル オブジェクトを返します。

フォレンジックプロファイルの削除

```
DELETE /openapi/v1/inventory_config/forensic_profiles/{profile_id}
```

パラメータ：

名前	タイプ	説明
profile_id	string	プロファイルの ID

成功すると 200 を返します。

フォレンジックルール

このソフトウェアエージェント API は、フォレンジックルールの管理に関連付けられています。

フォレンジックルールはフォレンジックプロファイルで使用され、エージェントのグループに適用されます。

これらの API のセットには、API キーに関連付けられている `'sensor_management'` 機能が必要です。

（注） これらの API は、サイト管理者とルート範囲の所有者のみが使用できます。

- フォレンジック ルール オブジェクト （1177 ページ）
- フォレンジックルールのリスト （1177 ページ）
- 単一のフォレンジックルールの取得 （1177 ページ）
- フォレンジックルールの作成 （1178 ページ）
- フォレンジックルールの更新 （1178 ページ）

• フォレンジックルールの削除 （1179 ページ）

フォレンジック ルール オブジェクト

属性	タイプ	説明
id	string	ルールの固有識別子
name	string	ルールの名前
説明	string	ルールの説明
type	string	PREDEFINED または USER_DEFINED
eval_group_type	string	AS_INDIVIDUAL または AS_GROUP
severity	string	IMMEDIATE_ACTION、CRITICAL、HIGH、MEDIUM、LOW のいずれか
アクション	array	配列または ALERT または REPORT の文字列
created_At	integer	ルールが作成されたときの Unix タイムスタンプ
updated_at	integer	ルールが最後に更新されたときの Unix タイムスタンプ

フォレンジックルールのリスト

このエンドポイントは、既存のすべてのフォレンジックルールをリストします。

```
GET /openapi/v1/inventory_config/forensic_rules
```

パラメータ：（なし）

このエンドポイントは、フォレンジック ルール オブジェクトの概要の配列を返します。

単一のフォレンジックルールの取得

```
GET /openapi/v1/inventory_config/forensic_rules/{rule_id}
```

パラメータ：

名前	タイプ	説明
rule_id	string	ルールの ID

フォレンジック ルール オブジェクトの詳細を返します。

フォレンジックルールの作成

```
POST /openapi/v1/inventory_config/forensic_rules
```

パラメータ：

名前	タイプ	説明
root_app_scope_id	string	このルールが属するルート範囲の ID
name	string	ルールの名前
説明	string	ルールの説明
eval_group_type	string	ルールのタイプ
severity	string	ルールの重大度
アクション	array	配列または ALERT または REPORT の文字列
clause	string	ルールのクエリ句

フォレンジック ルール オブジェクトを返します。

フォレンジックルールの更新

```
PUT /openapi/v1/inventory_config/forensic_rules/{rule_id}
```

パラメータ：

名前	タイプ	説明
rule_id	string	ルールの ID
name	string	ルールの名前
説明	string	ルールの説明
eval_group_type	string	ルールのタイプ
severity	string	ルールの重大度

名前	タイプ	説明
アクション	array	配列または ALERT または REPORT の文字列
clause	string	ルールのクエリ句

フォレンジック ルール オブジェクトを返します。

フォレンジックルールの削除

```
DELETE /openapi/v1/inventory_config/forensic_rules/{rule_id}
```

パラメータ：

名前	タイプ	説明
rule_id	string	ルールの ID

成功すると 200 を返します。

プラットフォーム設定

この一連の API は、プラットフォーム設定を追加、編集、または削除するために使用できます。API キーに関連付けられている appliance_management 機能が必要です。

（注） これらの API は、カスタマーサポートおよびサイト管理者のみが使用できます。

証明書の取得

このエンドポイントは、SSL/TLS 証明書を取得するために使用されます。

```
GET /openapi/v1/ platform_settings/ certs
```

応答は、次のスキーマを持つ JSON オブジェクトです。

キー	タイプ	値
key_pair_name	string	キー ペアの名前を指定します。
cert_sha1	string	証明書の固有識別子。
created_At	string	特定の日時以降に作成された証明書を取得します。

使用状況分析設定の取得

このエンドポイントは、プラットフォームでの使用状況分析またはテレメトリデータ収集に関連する設定を取得するために使用されます。

```
GET /openapi/v1/platform_settings/ usage_analytics
```

パラメータ：

名前	タイプ	説明
query	object	使用状況分析またはテレメトリデータ収集に関連する設定を取得します。

応答は、次のスキーマを持つ JSON オブジェクトです。

キー	タイプ	値
usage_analytics_enabled	boolean	使用状況分析またはテレメトリデータ収集が有効か無効かを示します。

ログインメッセージの取得

このエンドポイントは、プラットフォームのユーザーに表示されるカスタマイズされたログインメッセージを取得するために使用されます。

```
GET /openapi/v1/platfor_settings/ login_message
```

ログインメッセージの取得エンドポイントのパラメータはありません。

アウトバウンド HTTP 設定の取得

このエンドポイントは、プラットフォームに設定されている現在のアウトバウンドHTTP設定を取得するために使用されます。

```
GET /openapi/v1/platform_settings/outbound_http
```

応答は、次のスキーマを持つ JSON オブジェクトです。

キー	タイプ	値
outbound_http_enabled	string	アウトバウンド HTTP 接続が有効かどうかを示します。

アウトバウンド HTTP 設定の更新

このエンドポイントは、プラットフォームに設定されている現在のアウトバウンドHTTP設定を取得するために使用されます。

```
POST /openapi/v1/ platform_settings/ outbound_http
```

パラメータ：

名前	タイプ	説明
query	object	プラットフォームに設定されているアウトバウンド HTTP 設定を更新します。

応答は、次のスキーマを持つ JSON オブジェクトです。

キー	タイプ	値
`outbound_http_enabled`	string	アウトバウンド HTTP 接続が有効かどうかを示します。

アウトバウンド HTTP 設定のテスト

このエンドポイントは、プラットフォームに設定されている現在のアウトバウンドHTTP設定をテストするために使用されます。

```
POST openapi/v1/platform_settings/outbound_http_test
```

応答は、次のスキーマを持つ JSON オブジェクトです。

キー	タイプ	値
`success`	boolean	アウトバウンド HTTP 接続が成功したかどうかをテストします。

アウトバウンド HTTP プロキシ設定の取得

このエンドポイントは、プラットフォームに設定されているアウトバウンドHTTPプロキシ設定を取得するために使用されます。

```
GET /openapi/v1/platform_settings/outbound_http_proxy
```

応答は、次のスキーマを持つ JSON オブジェクトです。

キー	タイプ	値
outbound_http_enabled	boolean	アウトバウンド HTTP 接続が有効かどうかを示します。

キー	タイプ	値
http_proxy_enabled	boolean	HTTPプロキシが有効かどうかを示します。
http_proxy_server	string	設定されているアウトバウンド HTTP プロキシサーバーを取得します。
http_proxy_port	integer	HTTPプロキシサーバーに使用されるポート番号を指定します。
http_proxy_username	string	HTTPプロキシサーバーでの認証に使用されるユーザー名を指定します。
http_proxy_password_present	boolean	プラットフォームが HTTP プロキシサーバーでの認証にパスワードを使用するように現在設定されているかどうかを判断します。

アウトバウンド HTTP プロキシ設定の更新

このエンドポイントは、プラットフォームに設定されているアウトバウンドHTTPプロキシ設定を取得するために使用されます。

```
POST /openapi/v1/ platform_settings/ outbound_http_proxy
```

応答は、次のスキーマを持つ JSON オブジェクトです。

キー	タイプ	値
outbound_http_enabled	boolean	アウトバウンド HTTP 接続が有効かどうかを示します。
http_proxy_enabled	boolean	HTTPプロキシが有効かどうかを示します。
http_proxy_server	string	設定されているアウトバウンド HTTP プロキシサーバーを取得します。
http_proxy_port	integer	HTTPプロキシサーバーに使用されるポート番号を指定します。

キー	タイプ	値
http_proxy_username	string	HTTPプロキシサーバーでの認証に使用されるユーザー名を指定します。
http_proxy_password_present	boolean	パスワードが使用できるかどうかを決定します。

施行

ポリシーの適用は、生成されたポリシーがワークスペースに関連付けられた範囲内でアセットにプッシュされ、新しいファイアウォールルールが書き込まれる機能です。詳細については、「ポリシーの適用」マニュアルを参照してください。この一連の API には、API キーに関連付けられている `app_policy_management` 機能が必要です。

エージェント ネットワーク ポリシーの設定

このエンドポイントは、エージェント ID に従ってエージェントオブジェクトを返します。これは、ネットワーク ポリシー、エージェント設定、バージョンなどを取得するのに役立ちます。

```
GET /openapi/v1/enforcement/agents/{aid}/network_policy_config
```

パラメータ：

リクエスト URL には次のパラメータが含まれています

名前	タイプ	説明
aid	string	ネットワーク ポリシー設定のエージェント UUID。

JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
include_filter_names	boolean	ネットワークポリシーにフィルタ名と ID を含めます。
inject_versions	boolean	ネットワークポリシーにADMワークスペースバージョンを含めます。

応答

このエンドポイントの応答は、 エージェント オブジェクトです。

具体的なポリシーの統計情報

このエンドポイントは、エージェント ID と具体的なポリシー ID を指定して、具体的なポリシーの統計情報を返します。エンドポイントは、 Timeseries 具体的ポリシーの結果オブジェクトの配列を返します。

```
GET /openapi/v1/enforcement/agents/{aid}/concrete_policies/{cid}/stats?t0=<t0>&t1=<t1>
↪&td=<td>
```

パラメータ：

リクエスト URL には次のパラメータが含まれています

名前	タイプ	説明
aid	string	統計情報のエージェント UUID
CID	string	統計情報の具体的ポリシー UUID。

JSON クエリの本文には、次のキーが含まれています。

表 *60 :*

名前	タイプ	説明
t0	integer	エポック時間の統計情報の開始時刻
t1	integer	エポック時間の統計情報の終了時刻
td	整数または string	統計情報集約の粒度。整数で秒数を指定します。「分」、「時間」、「日」などの文字列をパスできます。

JSON オブジェクトの定義：

エージェント

属性	タイプ	説明
agent_uuid	string	エージェントの UUID。
agent_config	オブジェクト	エージェント設定
agent_config_status	オブジェクト	エージェント構成ステータス

属性	タイプ	説明
desired_network_policy_config	object	ネットワーク ポリシー設定
provisioned_network_policy_config	オブジェクト	プロビジョニングされたネットワークポリシー構成
provisioned_state_update_timestamp	integer	エージェントが上記のプロビジョニングされたポリシーを確認したときの秒単位のエポックタイムスタンプ。
desired_policy_update_timestamp	integer	desired_network_policy_config が生成されたときの秒単位のエポックタイムスタンプ。
agent_info	オブジェクト	エージェント情報
skipped	boolean	具体的ポリシーの生成がスキップされた場合は true。
message	string	具体的ポリシーの生成がスキップされた理由。

エージェント設定

属性	タイプ	説明
agent_uuid	string	エージェントの UUID。
enforcement_enabled	boolean	エージェントで適用が有効になっていることを示す構成。
fail_mode	string	フェールモード。
version	number	エージェント構成のバージョン番号。
control_tet_rules_only	boolean	tet ルールのみの構成を制御します。
allow_broadcast	boolean	ブロードキャスト構成を許可します。
allow_multicast	boolean	マルチキャスト構成を許可します。
allow_link_local	boolean	リンクローカル構成を許可します。

属性	タイプ	説明
enforcement_cpu_quota_mode	string	適用エージェントの CPU クォータモード。
enforcement_cpu_quota_us	string	適用エージェントの CPU クォータマイクロ秒。
enforcement_max_rss_limit	number	適用エージェントの最大 RSS 制限。

ネットワーク ポリシー設定

属性	タイプ	説明
version	string	バージョン番号。
network-policy	array	ネットワークポリシー オブジェクトの配列。
address_sets	array	IP セット機能のアドレス セットの配列。
container_network_policy	アレイ	ContainerNetworkPolicy オブジェクトの配列。

ネットワークポリシー

属性	タイプ	説明
priority	string	具体的ポリシーの優先順位。
enforcement_intent_id	string	適用インテント ID。
concrete_policy_id	string	具体的ポリシー ID。
match	オブジェクト	ポリシーの一致基準。これは廃止予定のフィールドです。
action	オブジェクト	ポリシー一致の操作（Action）。
workspace_id	string	ワークスペースの ID。
adm_data_set_id	string	ワークスペースの自動ポリシー検出データセット ID。

属性	タイプ	説明
adm_data_set_version	string	ワークスペースの自動ポリシー検出データセットバージョン。params に inject_versions=true が渡された場合にのみ設定します。
cluster_edge_id	string	クラスタエッジ ID。
policy_intent_group_id	string	ポリシーインテントグループ ID。
match_set	オブジェクト	IP セットサポートの一致セットオブジェクト。一致または match_set の 1 つだけが存在します。
src_filter_id	string	ソースインベントリフィルタ ID。これは、params として include_filter_names=true が渡されたときに設定されます。
src_filter_name	string	ソースインベントリフィルタ名。これは、params として include_filter_names=true が渡されたときに設定されます。
dst_filter_id	string	宛先インベントリフィルタ ID。これは、params として include_filter_names=true が渡されたときに設定されます。
dst_filter_name	string	宛先インベントリフィルタ名。これは、params として include_filter_names=true が渡されたときに設定されます。

ContainerNetworkPolicy

属性	タイプ	説明
pod_id	string	ポッド ID。
network-policy	array	ネットワークポリシー オブジェクトの配列。
deployment	string	展開名。

属性	タイプ	説明
service_endpoint	アレイ	サービスエンドポイント名のリスト。

一致

属性	タイプ	説明
src_addr	object	サブネットのサブネットオブジェクト。
dst-addr	object	サブネットのサブネットオブジェクト。
src_port_range_start	int	送信元ポートの範囲の開始。
src_port_range_end	int	送信元ポート範囲の終了。
dst_port_range_start	int	宛先ポート範囲の開始。
dst_port_range_end	int	宛先ポート範囲の終了。
ip_protocol	string	IP プロトコル。
address-family	string	IPv4 または IPv6 アドレス ファミリ。
direction	string	一致、入力、または出力の方向。
src_addr_range	object	送信元アドレスのアドレス範囲オブジェクト。
dst_add_range	object	宛先アドレスのアドレス範囲オブジェクト。

操作（Action）

属性	タイプ	説明
タイプ	string	アクションタイプ。

一致セット

属性	タイプ	説明
src_set_id	string	ネットワーク ポリシー設定 address_sets 配列のアドレス セットオブジェクトの送信元セット ID。 .
dst_set_id	string	ネットワーク ポリシー設定 address_sets 配列のアドレス セットオブジェクトの宛先セット ID。
src_port	ar- ray	送信元ポートのポート範囲オブジェクトの配列。
dst_ports	ar- ray	宛先ポートのポート範囲オブジェクトの配列。
ip_protocol	string	IP プロトコル。
address-family	string	IPv4 または IPv6 アドレス ファミリ。
direction	string	一致、入力、または出力の方向。

アドレス セット

属性	タイプ	説明
set-Id	string	アドレスセット ID。
addr_ranges	array	アドレス範囲オブジェクトの配列。
subnets	array	サブネットオブジェクトの配列。
addr_family	string	IPv4 または IPv6 アドレス ファミリ。

サブネット

属性	タイプ	説明
ip_addr	string	[IP Address]。
prefix_length	int	サブネットのプレフィックス長。

アドレス範囲

属性	タイプ	説明
start_ip_addr	string	範囲の開始 IP アドレス。
end_ip_addr	string	範囲の終了 IP アドレス。

ポート範囲

属性	タイプ	説明
start_port	int	範囲の開始ポート。
end_port	int	範囲の終了ポート。

エージェント構成ステータス

属性	タイプ	説明
無効	boolean	エージェントで適用が無効になっていることを示す構成。
current_version	number	エージェントに適用されているエージェント構成の現在のバージョン。
high_seen_version	number	エージェントが受信したエージェント構成の最新バージョン。

プロビジョニングされたネットワークポリシー構成

属性	タイプ	説明
version	string	エージェントによってプロビジョニングされたネットワークポリシー構成バージョン。

属性	タイプ	説明
error_reason	string	エージェントがポリシーを正常に適用した場合は CONFIG_SUCCESS、そうでない場合はエラーの理由。
無効	boolean	エージェントで適用が無効になっていることを示す構成。
current_version	number	エージェントに適用されている現在の NPC バージョン。
high_seen_version	number	エージェントが受信した NPC の最高バージョン。
policy_status	オブジェクト	すべてのネットワーク ポリシー ステータス。

エージェント情報

属性	タイプ	説明
agent_info_supported	boolean	エージェント機能で agent_info がサポートされているかどうか。
ipset_supported	boolean	エージェント機能で ipset がサポートされているかどうか。

具体的ポリシーの結果

属性	タイプ	説明
byte-count	int	具体的ポリシーヒットのバイト数。
pkt-count	int	具体的ポリシーヒットのパケット数。

Timeseries 具体的ポリシーの結果

属性	タイプ	説明
timestamp	string	結果集約のタイムスタンプ文字列。

属性	タイプ	説明
result	object	具体的ポリシーの結果

クライアントサーバー構成

クライアントとサーバーの関係の検出は、Secure Workloadのさまざまな機能の中核を成しており、グラウンドトゥルースを報告できるため、可能な場合は常にソフトウェアエージェントを使用することを推奨します。ネットワーク内のテレメトリ モニタリング ポイントでは、さまざまな状況により、特定のフローに関するすべてのパケットの監視は保証されません。たとえば、TCPフローにおける2つの単方向の半分がネットワークを通じて一意のパスを通過する可能性があるため、エラーのレベルにより常に影響を受けることを避けられません。

Cisco Secure Workloadは、各フローに機械学習アルゴリズムを適用し、一貫性のないテレメトリが報告された場合に判断を提供する統計モデルを構築することにより、ユーザーの操作なしでこれらのエラーを検出して最小限に抑えようとします。ほとんどの場合、ユーザーはこの一連のAPIについて気にする必要はありません。ただし、少数のケースでは、クライアントサーバーの検出アルゴリズムでフローの方向を正しく取得できません。たとえば、自動ポリシー検出など、フローの方向に依存する機能は、不要なポートを開くなどの望ましくない動作を示す場合があります。

既知のサーバーポートに関するヒントをSecure Workloadアルゴリズムに提供するために使用できる一連のAPIが用意されています。この一連のAPIは、ルート範囲の所有権ロールを持つユーザーが使用できます。それらのユーザーのAPIキーには、`app_policy_management`機能が関連付けられている必要があります。

クライアントサーバーの構成には2つのオプションがあります。

ホスト構成

ルート範囲内のIPアドレスの特定のサブセットに適用可能な既知のサーバーポートの構成

サーバーポート構成の追加

このAPIを使用して、特定のルート範囲の既知のサーバーポートに関するヒントをSecure Workloadアルゴリズムに提供できます。ルート範囲に属する一連のIPアドレスの既知のTCP/UDPサーバーポートのリストを提供して、フロー内のクライアントサーバーの正しい方向をSecure Workloadアルゴリズムに認識させることができます。

```
POST /openapi/v1/adm/{root_scope_id}/server_ports
```

パラメータ：要求URLには、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲の固有識別子。

さらに、このAPIへの入力として提供されるテキストファイルには、次の形式のエンドポイントサーバーのポート構成が含まれています。

エンドポイントサーバーのポート構成

属性	タイプ	説明
ip_address	string	IP アドレス（IPv4 または IPv6 アドレス）。サブネットは許可されていません。
tcp_server_ports	int のリスト	ip_address に対応する既知の TCP サーバーポートのリスト。
udp_server_ports	int のリスト	ip_address に対応する既知の UDP サーバーポートのリスト。

サーバーポートの一括設定

属性	タイプ	説明
host_config	エンドポイントサーバーのポート構成オブジェクトのリスト。	既知のサーバーポートが関連付けられている IP アドレスのリスト。

サンプル python コード

```
# contents of below file:
# {"host_config": [
#    {"ip_address": "1.1.1.1",
#     "tcp_server_ports": [100, 101, 102],
#     "udp_server_ports": [103]
#    },
#    {"ip_address": "1.1.1.2",
#     "tcp_server_ports": [200, 201, 202]
#    }
#  ]
# }

file_path = '/<path_to_file>/server_ports.txt'
root_scope_id = '<root-scope-id>'
restclient.upload(file_path,
                  '/adm/%s/server_ports' % root_scope_id,
                  timeout=200) # seconds
```

（注） 上記のAPIは、バックエンドの既知のサーバーポート構成の完全な状態を上書きします。変更が必要な場合は、変更後に完全な構成を再度アップロードする必要があります。

サーバーポート構成の取得

このAPIは、ルート範囲のエンドポイントに対してアップロードされた既知のサーバーポートのリストを返します。

```
GET /openapi/v1/adm/{root_scope_id}/server_ports
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲の固有識別子。

応答オブジェクト：ref:ServerPortConfig オブジェクトのリスト。

サンプル python コード

```
root_scope_id = '<root-scope-id>'
restclient.get('/adm/%s/server_ports' % root_scope_id)
```

サーバーポート構成の削除

この API は、指定されたルート範囲のサーバーポート構成を削除します。

```
DELETE /openapi/v1/adm/{root_scope_id}/server_ports_config
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲の固有識別子。

応答オブジェクト：なし。

サンプル python コード

```
root_scope_id = '<root-scope-id>'
restclient.delete('/adm/%s/server_ports' % root_scope_id)
```

ポート設定

ルート範囲に属するすべての IP アドレスに適用可能な既知のサーバーポートの設定

サーバーポート設定のプッシュ

このAPIを使用して、特定のルート範囲における既知のサーバーポートに関するヒントをSecure Workloadアルゴリズムに提供できます。ユーザーは、特定のルート範囲における既知のTCP/UDPサーバーポートのリストを提供して、フロー内のクライアントサーバーの正しい方向をSecure Workloadアルゴリズムに把握させることができます。また、各サーバーポートに関連付けられたサービス名を指定するオプションも使用できます。

すべてのルート範囲に適用可能な既知のサービスのデフォルトリストもあります（以降、グローバルサービスと呼びます）。このリストは、ユーザーがいつでも上書きできます。

サービス コンフィギュレーション

サービスは（ポート、名前の）ペアとして定義されます。

属性	タイプ	説明
port	int	TCP/UDPサーバーポート番号
name	string	このポートに関連付けられたサービス名（オプション）
override_in_conflicts	boolean	競合が発生した場合にホストを強制的にプロバイダーにする（オプション）

サービスの一括設定

属性	サブ属性	タイプ	説明
server_ports_config	tcp_service_list	サービス設定オブジェクトのリスト	既知の TCP サービスのリスト
	udp_service_list	サービス設定オブジェクトのリスト	既知の UDP サービスのリスト

```
Push services per root scope:

POST /openapi/v1/adm/{root_scope_id}/server_ports_config
```

サンプル python コード

```
# contents of below file:
#{"server_ports_config":
#    {
#        "tcp_service_list": [
#                {
#                        "port": 80,
#                        "name" : "http"
#
#                },
#                {
#                        "port": 53,
#                        "name" : "dns"
#                },
#                {
#                        "port": 514,
#                        "name" : "syslog",
#                        "override_in_conflicts": true
#                }
#        ],
#        "udp_service_list": [
#                {
#                        "port": 161
#                },
#                {
#                        "port": 53,
```

```
#                         "name": "dns"
#                 }
#         ]
#     }
#}

file_path = '/<path_to_file>/server_ports.json'

# Updating service list for a given root scope
#restclient.upload(file_path,
#                   '/openapi/v1/adm/{root_scope_id}/server_ports_config',
#                   timeout=200) # seconds
```



（注） 上記のAPIは、バックエンドの既知のサーバーポート設定の完全な状態を上書きします。ユーザーが何らかの変更を行う場合は、変更後に完全な設定を再度アップロードする必要があります。

サーバーポート設定の取得

このAPIは、ユーザーがアップロードしたルート範囲内における既知のサーバーポートのリストを返します。応答は一括サービス設定です。

```
Retrieve configured services per root scope:
    GET /openapi/v1/adm/{root_scope_id}/server_ports_config

Retrieve configured global services:
    GET /openapi/v1/adm/server_ports_config
```

サーバーポート設定の削除

この API は、指定されたルート範囲のサーバーポート設定を削除します。

```
Remove configured services per root scope:
    DELETE /openapi/v1/adm/{root_scope_id}/server_ports_config
```

ソフトウェアエージェント

エージェント API

ソフトウェアエージェントAPIは、Secure Workloadソフトウェアエージェントの管理に関連付けられています。これらのAPIのセットには、APIキーに関連付けられている `sensor_management` 機能が必要です。以下の GET API は、API キーに関連付けられた `flow_inventory_query` 機能でも使用できます。

ソフトウェア エージェントの取得

このエンドポイントは、ソフトウェア エージェントのリストを返します。

```
GET/openapi/v1/sensors
```

パラメータ：

名前	タイプ	説明
limit	integer	返される結果の数を制限します (オプション)
offset	string	オフセットはページネーション要求に使用されます。応答がオフセットを返す場合、後続の要求では同じオフセットを使用して、次のページでより多くの結果を取得する必要があります。（任意）

特定のソフトウェアエージェントの取得

このエンドポイントは、UUID が URI の一部であるエージェントの属性を返します。

```
GET /openapi/v1/sensors/{uuid}
```

ソフトウェア エージェントの削除

このエンドポイントは、UUIDを指定してソフトウェアエージェントを廃止するために使用されます。この API は注意して使用する必要があります。エージェントが削除されると、Secure Workload ダッシュボードに表示されなくなり、エージェントがアクティブな場合、Secure Workload ではエージェントからのフローエクスポートは許可されません。

```
DELETE /openapi/v1/sensors/{uuid}
```

インテントを使用したソフトウェア エージェントの設定

この API ワークフローは、以下に定義されているいくつかの REST エンドポイントを使用します。

インベントリ フィルタの作成

このエンドポイントは、ユーザーがソフトウェア エージェントを設定する、エージェント ホストに一致する基準を指定するために使用されます。

```
POST/openapi/v1/filters/inventories
```

パラメータ：

名前	タイプ	説明
app_scope_id	string	インベントリ フィルタに割り当てる範囲 ID。

名前	タイプ	説明
name	string	インベントリ フィルタの名前。
query	json	エージェント ホストのフィルタまたは一致基準。

サンプル python コード

```
# app_scope_id can be retrieved by /app_scopes API
req_payload = {
   "app_scope_id": <app_scope_id>,
   "name": "sensor_config_inventory_filter",
   "query": {
      "type": "eq",
      "field": "ip",
      "value": <sensor_interface_ip>
   }
}
resp = restclient.post('/filters/inventories',
                json_body=json.dumps(req_payload)))
print resp.status_code
# returned response will contain the created filter and it's ID.
```

ソフトウェア エージェント設定プロファイルの作成

このエンドポイントは、ソフトウェア エージェントのターゲット セットに適用する一連の設定オプションを指定するために使用されます。

`POST/openapi/v1/inventory_config/profiles`

次の設定オプションは、エージェント設定プロファイルの一部として指定できます。

- allow_broadcast：ブロードキャスト トラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- allow_multicast：マルチキャストトラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- allow_link_local：リンクローカルトラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- auto_upgrade_opt_out：True の場合、エージェントは Secure Workload クラスタのアップグレード中に自動アップグレードされません。
- cpu_quota_mode & cpu_quota_usec: これらのオプションは、エンド ホスト上のエージェントに割り当てる CPU クォータの量をポリシングするために使用されます。
- data_plane_disabled：True の場合、エージェントは Cisco Secure Workload へのフローのレポートを停止します。
- enable_conversation_mode：すべてのセンサーでカンバセーションモードを有効にするオプション。

- enable_forensics：ワークロードでのフォレンジックイベントの収集を有効にするオプション（結果的にエージェントはより多くの CPU を使用するようになります）。
- enable_meltdown：ワークロードでの Meltdown エクスプロイト検出を有効にします（結果的にエージェントはより多くの CPU を使用するようになります）。
- enable_pid_lookup: true の場合、エージェントはフローにプロセス情報を接続しようとします。(注) この設定オプションでは、エンド ホストでより多くの CPU が使用されます。
- enforcement_disabled: 適用エージェントを実行しているホストで適用を無効にするために使用できます。
- preserve_existing_rules: 既存の iptable ルールを保持するかどうかを指定するオプション。
- windows_enforcement_mode：WAF（Windows Advanced Firewall）または WFP（Windows Filtering Platform）を使用するオプション（デフォルトのオプションは WAF）。

設定オプションの詳細については、「ソフトウェアエージェントの設定」を参照してください。

サンプル python コード

```
# Define profile to disable data_plane on agent
req_payload = {
   "root_app_scope_id": <root_app_scope_id>,
   "data_plane_disabled": True,
   "name": "sensor_config_profile_1",
   "enable_pid_lookup": True,
   "enforcement_disabled": False
}
resp = restclient.post('/inventory_config/profiles',
                json_body=json.dumps(req_payload))
print resp.status_code
# returned response will contain the created profile and it's ID.
parsed_resp = json.loads(resp.content)
```

ソフトウェアエージェント設定プロファイルの取得

このエンドポイントは、ユーザーに表示されるソフトウェアエージェント設定プロファイルのリストを返します。

```
GET /openapi/v1/inventory_config/profiles
```

パラメータ：（なし）

特定のソフトウェアエージェント設定プロファイルの取得

このエンドポイントは、ソフトウェアエージェント設定プロファイルのインスタンスを返します。

```
GET /openapi/v1/inventory_config/profiles/{profile_id}
```

指定した ID に関連付けられているソフトウェアエージェント設定プロファイルオブジェクトを返します。

ソフトウェアエージェント設定プロファイルの更新

このエンドポイントは、ソフトウェアエージェント設定プロファイルを更新します。

```
GET /openapi/v1/inventory_config/profiles/{profile_id}
```

次の設定オプションは、エージェント設定プロファイルの一部として指定できます。

- allow_broadcast：ブロードキャスト トラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- allow_multicast：マルチキャストトラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- allow_link_local：リンクローカルトラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- auto_upgrade_opt_out：True の場合、エージェントは Secure Workload クラスタのアップグレード中に自動アップグレードされません。
- cpu_quota_mode & cpu_quota_usec: これらのオプションは、エンド ホスト上のエージェントに割り当てる CPU クォータの量をポリシングするために使用されます。
- data_plane_disabled：True の場合、エージェントは Cisco Secure Workload へのフローのレポートを停止します。
- enable_conversation_mode：すべてのセンサーでカンバセーションモードを有効にするオプション。
- enable_forensics：ワークロードでのフォレンジックイベントの収集を有効にするオプション（結果的にエージェントはより多くの CPU を使用するようになります）。
- enable_meltdown：ワークロードでの Meltdown エクスプロイト検出を有効にします（結果的にエージェントはより多くの CPU を使用するようになります）。
- enable_pid_lookup: true の場合、エージェントはフローにプロセス情報を接続しようとします。(注) この設定オプションでは、エンド ホストでより多くの CPU が使用されます。
- enforcement_disabled: 適用エージェントを実行しているホストで適用を無効にするために使用できます。
- preserve_existing_rules: 既存の iptable ルールを保持するかどうかを指定するオプション。
- windows_enforcement_mode：WAF（Windows Advanced Firewall）または WFP（Windows Filtering Platform）を使用するオプション（デフォルトのオプションは WAF）。

設定オプションの詳細については、「ソフトウェアエージェントの設定」を参照してください。

指定された ID に関連付けられている、変更されたソフトウェアエージェント設定プロファイルオブジェクトを返します。

ソフトウェアエージェント設定プロファイルの削除

このエンドポイントは、指定されたソフトウェアエージェント設定プロファイルを削除します。

```
DELETE /openapi/v1/inventory_config/profiles/{profile_id}
```

ソフトウェア エージェントの設定インテントの作成

このエンドポイントは、指定された一連のソフトウェアエージェントに一連の設定オプションを適用するインテントを指定するために使用されます。これにより、インテントが作成され、新しく作成されたインテントを順序に追加することによってインテントの順序が更新されます。

```
POST/openapi/v1/inventory_config/intents
```

サンプル python コード

```
req_payload = {
  "inventory_config_profile_id": <>,
  "inventory_filter_id": <>
}
resp = restclient.post('/inventory_config/intents',
               json_body=json.dumps(req_payload))
print resp.status_code
# returned response will contain the created intent object and it's ID.
```

インテントの順序の指定

このエンドポイントは、さまざまなソフトウェアエージェントの設定インテントの順序を指定するために使用されます。たとえば、2 つのインテントが存在する場合があります。一方は開発マシンのプロセス ID ルックアップを有効にし、他方は Windows マシンでのプロセス ID ルックアップを無効にします。最初のインテントの優先度が高い場合は、開発用の Windows マシンでのプロセス ID 検索が有効になります。注: デフォルトでは、インテントが作成されると、インテント順序リストの先頭に追加されます。このエンドポイントは、エンドユーザーが既存のインテントの順序を変更する必要がある場合にのみ使用されます。

```
POST/openapi/v1/inventory_config/orders
```

サンプル python コード

```
# Read the agent config intents ordered list
resp = restclient.get('/inventory_config/orders')
order_result_json = json.loads(resp.content)

# Modify the list by prepending the new intent in the list
order_rslt_json['intent_ids'].insert(0,<intent_id>)

# Post the new ordering back to the server
resp = restclient.post('/inventory_config/orders',
               json_body=json.dumps(order_rslt_json))
```

エージェント設定インテントの削除

このエンドポイントは、特定のエージェント設定インテントを削除するために使用されます。

```
DELETE /openapi/v1/inventory_config/intents/{intent_id}
```

サンプル python コード

```
intent_id = '588a51dcb5b30d0ee6da084a'
resp = restclient.delete('/inventory_config/intents/%s' % intent_id)
```

インターフェイス構成インテント

VRF をエージェントに割り当てるために推奨される方法は、リモート VRF 構成設定を使用することです。まれなケースですが、エージェントホストに複数のインターフェイスがあり、異なる VRF を割り当てる必要がある場合、ユーザーはインターフェイス構成インテントを使用してインターフェイスに VRF を割り当てることができます。**[管理（Manage）]** > **[エージェント（Agents）]** に移動し、**[設定（Configure）]** タブをクリックします。

Inventory Config Intent オブジェクト

GET および POST メソッドは、inventory config intent JSON オブジェクト配列を返します。オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
vrf-id	整数	VRF ID 整数
vrf_name	string	[VRF名（VRF Name）]
inventory_filter_id	string	インベントリフィルタ ID
inventory_filter	JSON	インベントリフィルタ。詳細については、OpenAPI > インベントリフィルタを参照してください。

インターフェイス構成インテントの取得

このエンドポイントは、ユーザーに表示されるインベントリ設定のリストを返します。

```
GET /openapi/v1/inventory_config/interface_intents
```

パラメータ：（なし）

インターフェイス構成インテントのリストの作成または更新

このエンドポイントは、インターフェイス構成インテントのリストを作成または変更するために使用します。API は、インテントの順序付きリストを受け取ります。このリストからインテントを削除するには、ユーザーは既存のインテントリストを読み込み、変更し、変更したリストを書き戻す必要があります。

```
POST /openapi/v1/inventory_config/interface_intents
```

パラメータ：

名前	タイプ	説明
inventory_filter_id	string	インターフェイスに一致するインベントリフィルタ ID
vrf-id	整数	インターフェイスを割り当てる VRF ID

サンプル python コード

```
  req_payload = {
    "intents": [
      {"inventory_filter_id": <inventory_filter_id_1>, "vrf_id": <vrf_id_1>},
      {"inventory_filter_id": <inventory_filter_id_1>, "vrf_id": <vrf_id_2>}
    ]
  }
  resp = restclient.post('/inventory_config/interface_intents',
json_body=json.dumps(req_payload))
```

NAT の背後にあるエージェントの VRF 設定

次の一連の API は、NAT ボックスの背後にあるエージェントに VRF を割り当てるポリシーを指定する場合に役立ちます。これらの API セットには、API キーに関連付けられている `sensor_management` 機能が必要であり、サイト管理ユーザーのみが使用できます。

NAT の背後にあるエージェントの VRF 設定ルールの一覧表示

このエンドポイントは、NAT の背後にあるエージェントに適用可能な VRF 設定ルールのリストを返します。

```
GET/openapi/v1/agentnatconfig
```

NAT の背後にあるエージェントに適用可能な新しい VRF 設定を作成します

このエンドポイントは、Secure Workload アプライアンスで認識される送信元 IP および送信元ポートに基づいて、ホストの VRF ラベル付け基準を指定するために使用されます。

```
POST/openapi/v1/agentnatconfig
```

パラメータ：

名前	タイプ	説明
src_subnet	string	送信元 IP が属することができるサブネット (CIDR 表記)。
src_port_range_start	整数	送信元ポート範囲の低バウンド (0-65535)。
src_port_range_end	整数	送信元ポート範囲の高バウンド (0-65535)。

名前	タイプ	説明
vrf-id	整数	送信元アドレスとポートが上記で指定した範囲内に収まるエージェントのフローのラベル付けに使用される VRF ID。

サンプル python コード

```
req_payload = {
    src_subnet: 10.1.1.0/24,          # src IP range for sensors
    src_port_range_start: 0,
    src_port_range_end: 65535,
    vrf_id: 676767                    # VRF ID to assign
}

resp = rc.post('/agentnatconfig', json_body=json.dumps(req_payload))
print resp.status_code
```

既存の VRF 設定の削除

```
DELETE /openapi/v1/agentnatconfig/{nat_config_id}
```

Cisco Secure Workload ソフトウェアのダウンロード

Secure Workload ソフトウェアのダウンロード機能では、Secure Workload エージェントのソフトウェアパッケージをダウンロードする方法を提供します。これらの一連の API には、API キーに関連付けられている `software_download` 機能が必要です。この機能は、サイト管理者ユーザー、ルート範囲の所有者、およびエージェントのインストーラロールを持つユーザーのみが使用できます。

サポートされているプラットフォームを取得するための API

このエンドポイントは、サポートされているプラットフォームのリストを返します。

```
GET /openapi/v1/sw_assets/platforms
```

パラメータ：（なし）

応答オブジェクト：サポートされているプラットフォームのリストを返します。

サンプル python コード

以下のサンプルコードは、サポートされているすべてのプラットフォームを取得します。

```
resp = restclient.get('/sw_assets/platforms')
if resp.status_code == 200:
   parsed_resp = json.loads(resp.content)
   print json.dumps(parsed_resp)
```

サンプル応答

```
{"results": [{"platform": "OracleServer-6.3", "agent_type": "enforcer", "arch": "x86_64"},
 {"platform": "MSWindows8Enterprise", "agent_type": "legacy_sensor", "arch": "x86_64"}]}
```

サポートされているソフトウェアバージョンを取得するための API

このエンドポイントは、指定された「agent_type」、「package_type」、「platform」、および「architecture」がサポートされているソフトウェアバージョンのリストを返します。

```
GET
/openapi/v1/sw_assets/download?platform=<platform>&agent_type=<agent_type>&pkg_type=<pkg_type>&arch=<arch>&list_version=<list_version>
```

このとき、<agent_type>、<platform>、<arch> は、**サポート対象のプラットフォームを取得するための API** から得た結果のいずれかにすることができます。<pkg_type> と <agent_type> はオプションですが、少なくともどちらか 1 つを指定する必要があります。この API を有効にするには、<list_version> が「True」である必要があります。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
platform	string	プラットフォームを指定します。
agent_type	string	（オプション）エージェントの種類を指定します。
pkg_type	string	（オプション）パッケージ タイプを指定します。値は「sensor_w_cfg」または「sensor_bin_pkg」のいずれかになります。
arch	string	アーキテクチャを指定します。
list_version	string	ソフトウェアバージョンの検索を有効にするには、「True」に設定します。

応答オブジェクト：サポートされているソフトウェアバージョンのリストを返します。

サンプル python コード

```
resp =
restclient.get('/sw_assets/download?platform=OracleServer-6.3&pkg_type=sensor_w_cfg&arch=x86_64&list_version=True')
if resp.status_code == 200:
   print resp.content
```

サンプル応答

```
3.3.1.30.devel
3.3.1.31.devel
```

インストーラ ID を作成する API

このエンドポイントは、Cisco Secure Workload ソフトウェアをダウンロードするための API のインストーラ ID を作成します。

```
GET /openapi/v1/sw_assets/installation_id
```

応答オブジェクト：Cisco Secure Workload ソフトウェアをダウンロードするために API で使用できるインストーラ ID を返します。

サンプル python コード

```
resp = restclient.get('/sw_assets/installation_id')
if resp.status_code == 200:
   parsed_resp = json.loads(resp.content)
   print json.dumps(parsed_resp)
```

サンプル応答

"[illegible]"

Secure Workload ソフトウェアをダウンロードするための API

このエンドポイントにより、クライアントは指定された「agent_type」、「package_type」、「platform」、「ar- chitecture」、および「sensor_version」のソフトウェアをダウンロードできます。

```
GET
/openapi/v1/sw_assets/download?platform=<platform>&agent_type=<agent_type>&pkg_type=<pkg_type>&arch=<arch>&sensor_version=<sensor_version>&installation_id=<installer_id>
```

このとき、<agent_type>、<platform>、<arch> は、**サポート対象のプラットフォームを取得するための API** から得た結果のいずれかにすることができます。<pkg_type> と <agent_type> はオプションですが、少なくともどちらか 1 つを指定する必要があります。<sensor_version> は、**サポート対象のソフトウェアバージョンを取得するための API** から得た結果のいずれかにできます。「sensor_version」が指定されていない場合は、**最新の**ソフトウェアがダウンロードされます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
platform	string	プラットフォームを指定します。
agent_type	string	（オプション）エージェントの種類を指定します。
pkg_type	string	（オプション）パッケージ タイプを指定します。値は「sensor_w_cfg」または「sensor_bin_pkg」のいずれかになります。

名前	タイプ	説明
arch	string	アーキテクチャを指定します。
sensor_version	string	（オプション）ソフトウェアのバージョンを指定します。デフォルトは空の文字列です。

応答オブジェクト：指定されたパラメータの Secure Workload ソフトウェアを返します。

サンプル python コード

```
resp =
if resp.status_code == 200:
   print 'file downloaded successfully'
```

Cisco Secure Workload エージェントのアップグレード

Secure Workload エージェントのアップグレード機能を使用して、インストールされている Secure Workload エージェントを特定のバージョンにアップグレードできます。この機能はメタデータのみを更新し、実際のアップグレードは次のチェックイン時に行われます。 API には、API キーに関連付けられた `software_download` 機能が必要です。この機能は、サイト管理者ユーザー、ルート範囲の所有者、またはエージェントのインストーラロールを持つユーザーのみが使用できます。

エージェントを特定のバージョンにアップグレードするための API

このエンドポイントは、「UUID」が特定の「sensor_version」にアップグレードされると、エージェントをトリガーします。「sensor_version」が指定されていない場合は、最新バージョンが適用されます。この API ではダウングレード要求は続行されません。

```
POST /openapi/v1/sensors/{UUID}/upgrade?sensor_version=<sensor_version>
```

<sensor_version> は、「 サポートされているソフトウェアバージョンを取得するための API 」から取得した結果のいずれかです。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
sensor_version	string	（オプション）希望のバージョンを指定します。デフォルトでは最新バージョンが適用されます。

このアップグレード要求のステータスを返します。

サンプル python コード

```
resp = restclient.post('/openapi/v1/sensors/{UUID}/upgrade?sensor_version=3.4.1.1.devel')

if resp.status_code == 200:
  print 'agent upgrade was triggered successfully and in progress'
elif resp.status_code == 304:
  print 'provided version is not newer than current version'
elif resp.status_code == 400:
  print 'provided version is invalid'
elif resp.status_code == 403:
  print 'user does not have required capability'
elif resp.status_code == 404:
  print 'agent with {UUID} does not exist'
```

ユーザーがアップロードしたファイルハッシュ

ユーザーはファイルハッシュのリストを Secure Workload にアップロードして、それらのハッシュが良性かフラグ付きかを指定できます。Secure Workloadはそれに応じて、それぞれのバイナリハッシュでプロセスにフラグを設定します。

この一連の API を使用して、ファイルハッシュのリストを Cisco Secure Workload にアップロードまたは削除できます。この API を呼び出すには、`user_data_upload` 機能で API キーを使用します。

（注） ルート範囲ごとに最大100万のファイルハッシュを持つことができます。良性ハッシュとフラグ付きハッシュについては、それぞれ 500000 のファイルハッシュを持つことができます。

次の API は、範囲所有者とサイト管理者が、「製品」アプライアンスの単一ルート範囲でファイルハッシュをアップロード、ダウンロード、および削除するために使用できます。

ユーザーファイルハッシュのアップロード

このエンドポイントは、Secure Workload アプライアンスのルート範囲のファイルハッシュを持つ CSV ファイルをアップロードする際に使用されます。列ヘッダーの `HashType` と `FileHash` が CSV ファイルに表示されている必要があります。`HashType` は `SHA-1` または `SHA-256` である必要があります。FileHash は空ではなく、40 文字で 16 進数の SHA1 形式または 64 文字で 16 進数の SHA256 形式である必要があります。

`FileName` および `Notes` ヘッダーはオプションです。ファイル名は 150 文字以下、注記は 1024 文字以下にする必要があります。

```
POST /openapi/v1/assets/user_filehash/upload/{rootAppScopeNameOrID}/{benignOrflagged}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeNameOrID	string	ルート範囲の名前または ID。

名前	タイプ	説明
benignOrflagged	string	benign と flagged のどちらか 1 つ。

応答オブジェクト：なし

サンプル python コード

```
  # Sample CSV File
  # HashType,FileHash,FileName,Notes
  # SHA-1,1AF17E73721DBE0C40011B82ED4BB1A7DBE3CE29,application_1.exe,Sample Notes
  #
SHA-256,8F434346648F6B96DF89DDA901C5176B10A6D83961DD3C1AC88B59B2DC327AA4,application_2.exe,Sample
 Notes

  file_path = '/<path_to_file>/user_filehash.csv'
  root_app_scope_name = 'Tetration'
  restclient.upload(file_path, '/assets/user_filehash/upload/%s/benign' %
root_app_scope_name)
```

ユーザーファイルハッシュの削除

このエンドポイントは、CSV ファイルをアップロードして、Secure Workload アプライアンスのルート範囲からファイルハッシュを削除するために使用されます。CSV ファイルには、ヘッダーとして FileHash が必要です。

```
POST /openapi/v1/assets/user_filehash/delete/{rootAppScopeNameOrID}/{benignOrflagged}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeNameOrID	string	ルート範囲の名前または ID。
benignOrflagged	string	benign と flagged のどちらか 1 つ。

応答オブジェクト：なし

サンプル python コード

```
  # Sample CSV File
  # FileHash
  # 1AF17E73721DBE0C40011B82ED4BB1A7DBE3CE29
  # 8F434346648F6B96DF89DDA901C5176B10A6D83961DD3C1AC88B59B2DC327AA4

  file_path = '/<path_to_file>/user_filehash.csv'
  root_app_scope_name = 'Tetration'
  restclient.upload(file_path, '/assets/user_filehash/delete/' + root_app_scope_name +
'/benign')
```

ユーザーファイルハッシュのダウンロード

このエンドポイントは、Secure Workload アプライアンスの指定されたルート範囲のユーザーファイルハッシュを CSV ファイルとして返します。CSV ファイルには、ヘッダー `HashType`、`FileHash`、`FileName`、および `Notes` がそれぞれの順序で含まれます。

```
GET /openapi/v1/assets/user_filehash/download/{rootAppScopeNameOrID}/{benignOrflagged}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeNameOrID	string	ルート範囲の名前または ID。
benignOrflagged	string	良性またはフラグ付きのいずれかです。

応答オブジェクト：なし

サンプル python コード

```
  file_path = '/<path_to_file>/output_user_filehash.csv'
  root_app_scope_name = 'Tetration'
  restclient.download(file_path, '/assets/user_filehash/download/%s/benign' %
root_app_scope_name)
```

ユーザー定義ラベル

この API は、Secure Workload アプライアンスのフローとインベントリ項目にラベル付けをするためのユーザー定義ラベルを追加または削除するために使用します。この API を呼び出すには、`user_data_upload` 機能で API キーを使用します。フローとインベントリ項目のラベル付けに使用するキーと値の管理に関するガイドラインについては、UI ユーザーガイドの「ラベルキースキーマ」セクションを参照してください。

（注） この機能に UI を使用してアクセスする方法については、「カスタムラベルのインポート」を参照してください。

（注） アップロードできる IPv4/IPv6 アドレスやサブネットの数に関する制限事項については、「ラベルの制限」を参照してください。

範囲依存 API

次の API は、Secure Workload アプライアンスの単一のルート範囲でラベルを取得/設定/削除するために使用されます。これらは、ルート**範囲の所有者**と**サイト管理者**のみが使用できます。さらに、ルート範囲への**読み取りアクセス権**を持つユーザーは、GET API呼び出しを使用できます。

インベントリラベルの取得

このエンドポイントは、Secure Workload アプライアンスのルート範囲にある IPv4/IPv6 アドレスまたはサブネットのラベルを返します。このエンドポイントのクエリに使用されるアドレス/サブネットは、ラベルのアップロードに使用されるものと完全に一致している必要があります。

```
GET /openapi/v1/inventory/tags/{rootAppScopeName}?ip={IPorSubnet}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。
IPorSubnet	string	IPv4/IPv6 アドレスまたはサブネット。

応答オブジェクト：

名前	タイプ	説明
attributes	JSON	フローとインベントリの各項目のマッチングをラベリングするためのキー/値マップ。

サンプル python コード

```
root_app_scope_name = 'Tetration'
restclient.get('/inventory/tags/%s' % root_app_scope_name, params={'ip': '10.1.1.1/24'})
```

インベントリラベルの検索

このエンドポイントでは、Secure Workload アプライアンスのルート範囲で IPv4/IPv6 アドレスまたはサブネットのラベルを検索できます。

```
GET /openapi/v1/inventory/tags/{rootAppScopeName}/search?ip={IPorSubnet}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

名前	タイプ	説明
IPorSubnet	string	IPv4/IPv6 アドレスまたはサブネット。

応答オブジェクト：この API は、次の形式でオブジェクトのリストを返します。

名前	タイプ	説明
キー	string	IPv4/IPv6 アドレスまたはサブネット。
updatedAt	integer	ラベルが更新されたときの UNIX タイムスタンプ 。
value	JSON	キーの属性のキー/値マップ。

サンプル python コード

```
  root_app_scope_name = 'Tetration Scope'
  encoded_root_app_scope_name = urllib.quote(root_app_scope_name, safe='')
  restclient.get('/inventory/tags/%s/search' % encoded_root_app_scope_name, params={'ip':
 '10.1.1.1/24'})
```

インベントリラベルの設定

このエンドポイントは、Secure Workload アプライアンスのルート範囲でフローとインベントリの各項目にラベルを付けるためのラベルの設定に使用されます。

```
POST /openapi/v1/inventory/tags/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
ip	string	IPv4/IPv6 アドレスまたはサブネット。
attributes	JSON	フローとインベントリの各項目のマッチングをラベリングするためのキー/値マップ。

応答オブジェクト：

名前	タイプ	説明
警告	JSON	ラベルの設定中に発生した警告を含むキー/値マップ。

サンプル python コード

```
  root_app_scope_name = 'Tetration'
  req_payload = {'ip': '10.1.1.1/24', 'attributes': {'datacenter': 'SJC', 'location':
'CA'}}
  restclient.post('/inventory/tags/%s' % root_app_scope_name,
json_body=json.dumps(req_payload))
```

インベントリラベルの削除

このエンドポイントは、Secure Workload アプライアンスのルート範囲にある IPv4/IPv6 アドレスまたはサブネットのラベルを削除します。

```
DELETE /openapi/v1/inventory/tags/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
ip	string	IPv4/IPv6 アドレスまたはサブネット。

サンプル python コード

```
  root_app_scope_name = 'Tetration'
  req_payload = {'ip': '10.1.1.1/24'}
  restclient.delete('/inventory/tags/%s' % root_app_scope_name,
json_body=json.dumps(req_payload))
```

ラベルのアップロード

このエンドポイントは、Secure Workload アプライアンスのルート範囲でフローとインベントリの各項目にラベルを設定するためのラベル付きの CSV ファイルをアップロードするために使用されます。`IP` という名前の列ヘッダーが CSV ファイルに表示されている必要があります。残りの列ヘッダーのうち、最大 32 までを使用して、フローとインベントリ項目の注釈を付けることができます。ラベルに英語以外の文字を使用するには、アップロードした CSV ファイルが UTF-8 形式である必要があります。

```
POST /openapi/v1/assets/cmdb/upload/{rootAppScopeName}
```

パラメータ：

ユーザーは、この API へのパラメータとして操作タイプ（X-Tetration-Oper）を提供する必要があります。X-Tetration-Oper は、次のいずれかとすることができます。

- [追加（add）]：ラベルを新規および既存のアドレス/サブネットに追加します。既存のラベルの代わりに新しいラベルを選択して、競合を解決します。たとえば、データベース内の住所のラベルが *{"foo": "1", "bar": "2"}* で、CSV ファイルに *{"z": "1", "bar": "3"}* が含まれている場合、*add* は、このアドレスのラベルを *{"foo": "1", "z": "1", "bar": "3"}* に設定します。
- [上書き（overwrite）]：新しいアドレス/サブネットのラベルを挿入し、既存のラベルを置き換えます。たとえば、データベース内の住所のラベルが *{"foo": "1", "bar": "2"}* で、CSV ファイルに *{"z": "1", "bar": "3"}* が含まれている場合、*overwrite* は、このアドレスのラベルを *{“z”: “1”, “bar”: “3”}* に設定します。
- [マージ（merge）]：ラベルを既存のアドレス/サブネットにマージします。空の値の代わりに空でない値を選択することで、競合を解決します。たとえば、データベース内のアドレスのラベルが *{"foo": "1", "bar": "2", "qux": "", "corge": "4"}* で、CSV ファイルに *{"z": "1", "bar": "", "qux": "3", "corge": "4-updated"}* が含まれている場合、*merge* は、このアドレスのラベルを *{"foo": "1", "z": "1", "bar": "2", "qux": "3", "corge": "4-updated"}* に設定します。

（注） “bar” の値は “”（空）にリセットされず、既存の値 “bar”=”2” が保持されます。

- [削除（delete）]：アドレス/サブネットのラベルを削除します。

応答オブジェクト：

名前	タイプ	説明
警告	JSON	ラベルの設定中に発生した警告を含むキー/値マップ。

サンプル python コード

```
file_path = '/<path_to_file>/user_annotations.csv'
root_app_scope_name = 'Tetration'
req_payload = [tetpyclient.MultiPartOption(key='X-Tetration-Oper', val='add')]
restclient.upload(file_path, '/assets/cmdb/upload/%s' % root_app_scope_name, req_payload)
```

ユーザーラベルのダウンロード

このエンドポイントは、Secure Workload アプライアンスのルート範囲に対してユーザーがアップロードしたラベルを CSV ファイルとして返します。

```
GET /openapi/v1/assets/cmdb/download/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答:

Content-Type: *text/csv*

ユーザーが範囲に対してアップロードしたラベルを含む CSV ファイル。

サンプル python コード

```
file_path = '/<path_to_file>/output.csv'
root_app_scope_name = 'Tetration'
restclient.download(file_path, '/assets/cmdb/download/%s' % root_app_scope_name)
```

列ヘッダーの取得

このエンドポイントは、Secure Workload アプライアンスのルート範囲の列ヘッダーのリストを返します。

```
GET /openapi/v1/assets/cmdb/attributenames/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答オブジェクト：ラベルに使用できるファセットの配列。

サンプル python コード

```
root_app_scope_name = 'Tetration'
resp = restclient.get('/assets/cmdb/attributenames/%s' % root_app_scope_name)
```

列ヘッダーの削除

このエンドポイントは、Secure Workload アプライアンスのルート範囲の列ヘッダーを削除します。列ヘッダーを削除すると、ラベル付きファセットのリストから削除され、既存のラベルからも削除されます。

```
DELETE /openapi/v1/assets/cmdb/attributenames/{rootAppScopeName}/{attributeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。
attributeName	string	削除対象の属性。

応答オブジェクト：なし

サンプル python コード

```
   root_app_scope_name = 'Tetration'
   attribute_name = 'column1'
   resp = restclient.delete('/assets/cmdb/attributenames/%s/%s' % (root_app_scope_name,
 attribute_name))
```

ラベル付きファセットのリストの取得

このエンドポイントは、Secure Workloadアプライアンスのルート範囲のラベル付きファセットのリストを返します。ラベル付きファセットは、その範囲内のフローとインベントリ項目に注釈を付けるために使用される、アップロードされたCSVファイル内の列ヘッダーのサブセットです。

```
GET /openapi/v1/assets/cmdb/annotations/{rootAppScopeName}
```

パラメータ：要求URLには、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答オブジェクト：ルート範囲のラベル付きファセットの配列。

サンプル python コード

```
   root_app_scope_name = 'Tetration'
   resp = restclient.get('/assets/cmdb/annotations/%s' % root_app_scope_name)
```

ラベル付きファセットのリストの更新

このエンドポイントは、Secure Workloadアプライアンスのルート範囲のフローとインベントリ項目に注釈を付けるために使用されるファセットのリストを更新します。

```
PUT /openapi/v1/assets/cmdb/annotations/{rootAppScopeName}
```

パラメータ：要求URLには、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答オブジェクト：なし

サンプル python コード

```
   # the following list is a subset of column headers in the
   # uploaded CSV file
   req_payload = ['location', 'region', 'detail']
   root_app_scope_name = 'Tetration'
   restclient.put('/assets/cmdb/annotations/%s' % root_app_scope_name,
                  json_body=json.dumps(req_payload))
```

ユーザーがアップロードしたラベルの消去

このエンドポイントは、Secure Workloadアプライアンスのルート範囲でフローとインベントリの各項目のラベルを消去します。変更は新しいデータに影響します。古いラベル付きデータは変更されずに残ります。

```
POST /openapi/v1/assets/cmdb/flush/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答オブジェクト：なし

サンプル python コード

```
root_app_scope_name = 'Tetration'
restclient.post('/assets/cmdb/flush/%s' % root_app_scope_name)
```

範囲に依存しない API

次の API は、Secure Workload アプライアンス上の複数の範囲にまたがる場合があります。

（注） 範囲非依存および範囲依存の注釈付きファセットの数は、どのルート範囲でも 32 を超えてはなりません。

ラベルのアップロード

このエンドポイントは、Secure Workloadアプライアンスのフローとインベントリの各項目にラベルを設定するためのラベル付きの CSV ファイルをアップロードするために使用されます。`IP` および `VRF` という名前の列ヘッダーが CSV ファイルに表示され、`VRF` がラベルのルート範囲と一致する必要があります。残りの列ヘッダーのうち、最大 32 までを使用して、フローとインベントリ項目の注釈を付けることができます。

```
POST/openapi/v1/assets/cmdb/upload
```

パラメータ：

ユーザーは、実行する操作を指定するために、この API のパラメータとして操作タイプ（`X-Tetration-Oper`）を提供する必要があります。

応答オブジェクト：

名前	タイプ	説明
警告	JSON	ラベルの設定中に発生した警告を含むキー/値マップ。

サンプル python コード

```
file_path = '/<path_to_file>/user_annotations.csv'
req_payload = [tetpyclient.MultiPartOption(key='X-Tetration-Oper', val='add')]
restclient.upload(file_path, '/assets/cmdb/upload', req_payload)
```

ユーザーラベルのダウンロード

このエンドポイントは、Secure Workload アプライアンスのすべての範囲に対してユーザーがアップロードしたラベルを CSV ファイルとして返します。

```
GET/openapi/v1/assets/cmdb/download
```

パラメータ：（なし）

応答:

Content-Type: *text/csv*

ユーザーが範囲に対してアップロードしたラベルを含む CSV ファイル。

サンプル python コード

```
file_path = '/<path_to_file>/output.csv'
restclient.download(file_path, '/assets/cmdb/download')
```

範囲に依存しないラベル

これらのラベルは、特定のルート範囲に関連付けられておらず、アプライアンスのすべての範囲に適用されます。

インベントリラベルの取得

このエンドポイントは、Secure Workload アプライアンスの IPv4/IPv6 アドレスまたはサブネットの範囲に依存しないラベルを返します。このエンドポイントのクエリに使用されるアドレス/サブネットは、ラベルのアップロードに使用されるものと完全に一致している必要があります。

```
GET /openapi/v1/si_inventory/tags?ip={IPorSubnet}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
IPorSubnet	string	IPv4/IPv6 アドレスまたはサブネット。

応答オブジェクト：

名前	タイプ	説明
attributes	JSON	フローとインベントリの各項目のマッチングをラベリングするためのキー/値マップ。

サンプル python コード

```
restclient.get('/si_inventory/tags', params={'ip': '10.1.1.1/24'})
```

インベントリラベルの検索

このエンドポイントにより、Secure Workload アプライアンスの IPv4/IPv6 アドレスまたはサブネットのラベルを検索できます。

```
GET /openapi/v1/si_inventory/tags/search?ip={IPorSubnet}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
IPorSubnet	string	IPv4/IPv6 アドレスまたはサブネット。

応答オブジェクト：この API は、次の形式でオブジェクトのリストを返します。

名前	タイプ	説明
キー	string	IPv4/IPv6 アドレスまたはサブネット。
updatedAt	integer	ラベルが更新されたときの UNIX タイムスタンプ 。
value	JSON	キーの属性のキー/値マップ。

サンプル python コード

```
restclient.get('/si_inventory/tags/search', params={'ip': '10.1.1.1/24'})
```

インベントリラベルの設定

このエンドポイントは、Secure Workload アプライアンスのフローとインベントリの各項目にラベルを付けるためのラベルの設定に使用されます。

```
POST /openapi/v1/si_inventory/tags
```

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
ip	string	IPv4/IPv6 アドレスまたはサブネット。
attributes	JSON	フローとインベントリの各項目のマッチングをラベリングするためのキー/値マップ。

応答オブジェクト：

名前	タイプ	説明
警告	JSON	ラベルの設定中に発生した警告を含むキー/値マップ。

サンプル python コード

```
  req_payload = {'ip': '10.1.1.1/24', 'attributes': {'datacenter': 'SJC', 'location':
'CA'}}
  restclient.post('/si_inventory/tags', json_body=json.dumps(req_payload))
```

インベントリラベルの削除

このエンドポイントは、Secure Workload アプライアンスの IPv4/IPv6 アドレスまたはサブネットのラベルを削除します。

```
DELETE /openapi/v1/si_inventory/tags
```

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
ip	string	IPv4/IPv6 アドレスまたはサブネット。

サンプル python コード

```
  req_payload = {'ip': '10.1.1.1/24'}
  restclient.delete('/si_inventory/tags, json_body=json.dumps(req_payload))
```

ラベル付きファセットのリストの取得

このエンドポイントは、Secure Workload アプライアンス上の範囲に依存しないラベル付きファセットのリストを返します。ラベル付きファセットは、すべての範囲のフローおよびインベントリ項目にラベルを付けるために使用される列ヘッダーのサブセットです。

（注） 要求 URL から範囲名を除外して、注釈付き範囲に依存しないファセットのリストを表示および更新します。

```
GET /openapi/v1/assets/cmdb/annotations
```

応答オブジェクト：範囲に依存しないラベル付きファセットの配列。

サンプル python コード

```
resp = restclient.get('/assets/cmdb/annotations')
```

ラベル付きファセットのリストの更新

このエンドポイントは、Secure Workloadアプライアンス上のフローおよびインベントリ項目に注釈を付けるために使用される、範囲に依存しないファセットのリストを更新します。

```
PUT /openapi/v1/assets/cmdb/annotations
```

応答オブジェクト：なし

サンプル python コード

```
# the following list is a subset of column headers in the
# uploaded CSV file
req_payload = ['location', 'region', 'detail']
restclient.put('/assets/cmdb/annotations',
               json_body=json.dumps(req_payload))
```

Virtual Routing and Forwarding（VRF）

この API のセットは、VRF を管理します。

（注） これらの API は、サイト管理者のみ使用できます。

VRF オブジェクト

VRF オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	int	VRF の固有識別子。
name	string	ユーザーが指定した VRF の名前。
tenant_Id	int	親テナントの ID。
root_app_scope_id	string	関連付けられているルート範囲の ID。
created_At	integer	VRF が作成されたときの Unix タイムスタンプ。

属性	タイプ	説明
updated_at	integer	VRF が最後に更新されたときの Unix タイムスタンプ。

VRF の取得

このエンドポイントは、VRF のリストを返します。この API は、`sensor_management` または `flow_inventory_query` 機能を備えた API キーで使用できます。

`GET/openapi/v1/vrfs`

パラメータ：（なし）

応答オブジェクト：VRF オブジェクトのリストを返します。

サンプル python コード

```
resp = restclient.get('/vrfs')
```

VRF を作成します

このエンドポイントは、新しい VRF を作成するために使用されます。関連付けられたルート範囲は、VRF ID に一致するクエリを使用して自動的に作成されます。この API は、`sensor_management` 機能を持つ API キーで使用できます。

`POST/openapi/v1/vrfs`

パラメータ：

名前	タイプ	説明
id	int	（オプション）VRF の固有識別子。指定しない場合、Secure Workload クラスタは新しく作成された VRF の一意の ID を生成します。ベストプラクティスは、呼び出し元が一意の ID を明示的に指定する代わりに、Cisco Secure Workload にこれらの ID を生成させることです。
tenant_Id	int	（オプション）親テナントの ID。
name	string	ユーザーが指定した VRF の名前。

名前	タイプ	説明
apply_monitoring_rules	boolean	（オプション）収集ルールを VRF に適用するかどうかを指定します。デフォルトは「false」です。

tenant_id はオプションです。指定しない場合、VRF は VRF と同じ ID を持つテナント（必要に応じて自動作成される）に追加されます。tenant_id が指定されている場合、テナントは自動作成されず、テナントが存在しない場合はエラーが返されます。

応答オブジェクト：新しく作成された VRF オブジェクトを返します。

サンプル python コード

```
req_payload = {
    "tenant_id": <tenant_id>,
    "name": "Test",
    "apply_monitoring_rules": True
}
resp = restclient.post('/vrfs', json_body=json.dumps(req_payload))
```

特定の VRF の取得

このエンドポイントは、指定された VRF ID の情報を返します。この API は、sensor_management または flow_inventory_query 機能を備えた API キーで使用できます。

```
GET /openapi/v1/vrfs/{vrf_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
vrf-id	int	VRF の固有識別子。

応答オブジェクト：指定された ID に関連付けられている VRF オブジェクトを返します。

サンプル python コード

```
vrf_id = 676767
resp = restclient.get('/vrfs/%d'% vrf_id)
```

VRF の更新

このエンドポイントは、VRF を更新します。この API は、sensor_management 機能を備えた API キーで使用できます。

```
PUT/openapi/v1/vrfs/{vrf_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
vrf-id	int	VRF の固有識別子。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
name	string	ユーザーが指定した VRF の名前。
apply_monitoring_rules	boolean	（オプション）収集ルールを VRF に適用するかどうかを指定します。

応答オブジェクト：指定された ID に関連付けられている変更された VRF オブジェクトを返します。

サンプル python コード

```
vrf_id = 676767
req_payload = {
    "name": "Test",
    "apply_monitoring_rules": True
}
resp = restclient.put('/vrfs/%d'% vrf_id,
                      json_body=json.dumps(req_payload))
```

特定の VRF の削除

このエンドポイントは、VRF を削除します。ルート範囲が関連付けられている場合、これは失敗します。この API は、`sensor_management` 機能を備えた API キーで使用できます。

```
DELETE /openapi/v1/vrfs/{vrf_id}
```

パラメータ：次のパラメータは URL の一部です

名前	タイプ	説明
vrf-id	int	VRF の固有識別子。

サンプル python コード

```
vrf_id = 676767
resp = restclient.delete('/vrfs/%d'% vrf_id)
```

オーケストレーション

この一連の API を使用して、Secure Workload クラスタ展開での外部オーケストレータのインベントリ学習を管理できます。API キーに関連付けられた `external_integration` 機能が必要です。

現在サポートされているオーケストレーションタイプは、「vcenter」（VCenter 6.5 以降）、「kubernetes」、「dns」、「f5」、「netscaler」、「infoblox」および「Cisco FMC」です。外部オーケストレータにあるサポートされるユーザーインターフェイス。

オーケストレーション オブジェクト

オーケストレータ オブジェクトの属性について次に説明します。一部のフィールドは、特定のオーケストレータタイプにのみ適用されます。以下の表に制限事項を示します。

属性	タイプ	説明
id	string	オーケストレーションの固有識別子。
name	string	ユーザーが指定したオーケストレーションの名前。
type	string	オーケストレータのタイプ：サポートされる値（vcenter、kubernetes、f5、netscaler、infoblox、dns）
説明	string	オーケストレーションのユーザー指定の説明。
username	string	オーケストレーション エンドポイントのユーザー名。（dns の場合は不要）
password	string	オーケストレーション エンドポイントのパスワード。（dns の場合は不要）
証明書	string	認証に使用されるクライアント証明書（dns の場合は不要）
キー	string	クライアント証明書に対応するキー（dns の場合は不要）

属性	タイプ	説明
ca_certificate	string	オーケストレーション エンドポイントを検証する CA 証明書（dns の場合は不要）
auth_token	string	不透明な認証トークン（ベアラートークン）（kubernetes にのみ適用）
insecure	boolean	厳密な SSL 検証を無効にする
delta_interval	整数	秒単位のデルタポーリング間隔の Secure Workload Inventory Manager は、delta_interval 秒ごとに増分変更のポーリングを実行します。このパラメータは、Infoblox および Cisco Secure Firewall Management Center には適用されないことに注意してください。
full_snapshot_interval	整数	秒単位のフルスナップショット間隔の Secure Workload Inventory Manager は、オーケストレータからフル更新ポーリングを実行します。
verbose_tsdb_metrics	boolean	エンドポイントごとの TSDB メトリックス
hosts_list	Array	オーケストレータへの Secure Workload の接続方法を指定する { “host_name”, port_number} ペアの配列
use_secureconnector_tunnel	boolean	Secure Connector トンネルを介したこのオーケストレータのホストへのトンネル接続
route_domain	integer	F5 ロードバランサでポーリングするルートドメイン番号（「f5」にのみ適用）

属性	タイプ	説明
dns_zones	配列	DNS サーバーからポーリングする DNS ゾーンを含む文字列の配列（ dns のみ）。各 DNS ゾーンエントリは、a で終わる必要があります。
enable_enforcement	boolean	ファイアウォールやロードバランサなど、ポリシーの適用をサポートする外部オーケストレータにのみ適用されます。たとえば、Cisco Secure Firewall Management Center、F5 BIGIP、および Citrix Netscaler などがあります。デフォルトでは、このフラグは false（ポリシーの強制は無効）です。true の場合、ワークスペースに対してポリシーの適用が実行されると、外部オーケストレータは指定されたロードバランサ アプライアンスにポリシーを展開します。
ingress_controllers	オブジェクト	Ingress コントローラ オブジェクトの配列。
fmc_enforcement_mode	string	Cisco Secure Firewall Management Center 外部オーケストレータにのみ適用され、マージ（デフォルト）またはオーバーライドのいずれかである必要があります。最初のインスタンスは、既存のプレフィルタルールより先にすべての Secure Workload ポリシールールを配置するように、Cisco Secure Firewall Management Center ポリシーエンフォーサに指示します。後者のインスタンスは、ユーザーによって作成されたすべてのプレフィルタルールを削除します。

属性	タイプ	説明
infoblox_config	オブジェクト	Infoblox外部オーケストレータにのみ適用されます。Infoblox設定 レコードタイプセレクタ。

Ingress コントローラ

属性	タイプ	説明
pod_selector	オブジェクト	ポッドセレクタ
controller_config	オブジェクト	コントローラ設定

ポッドセレクタ

属性	タイプ	説明
名前空間	string	Ingress コントローラポッドが実行されている名前空間。
ラベル	配列	Ingress コントローラポッドのラベルを指定する {“key”, “value”} ペアの配列。

コントローラ設定

属性	タイプ	説明
ingress_class	string	Ingress コントローラが満たす入力クラスの名前。
名前空間	string	名前空間は、Ingress コントローラが満たす名前空間の名前です。
http_ports	配列	HTTP ポートの配列。
https_ports	配列	HTTPS ポートの配列。

Infoblox 設定

enable_network_record	bool	デフォルト値はtrueです。falseの場合、ネットワークタイプのレコードは無効になります。
enable_host_record	bool	デフォルト値はtrueです。falseの場合、ホストタイプのレコードは無効になります。
enable_a_record	bool	デフォルト値はtrueです。falseの場合、*A*タイプのレコードは無効になります。
enable_aaaa_record	bool	デフォルト値はtrueです。falseの場合、*AAAA*タイプのレコードは無効になります。

** オーケストレータオブジェクトの読み取り専用ステータスフィールド **

属性	タイプ	説明
authentication_failure	bool	Secure Workload オーケストレータへの接続ステータス：*true* はオーケストレータへの接続が成功したことを示します。このフィールドが *false* の場合、*authentication_failure_error* フィールドには、接続失敗の理由を説明する詳細なエラーメッセージが表示されます。
authentication_failure_error	string	オーケストレータとの接続エラーやクレデンシャルエラーのデバッグに役立つ詳細なエラーメッセージ
scope_id	string	インベントリが公開および表示されるテナントルート範囲ID

オーケストレーションの取得

このエンドポイントは、Secure Workload アプライアンスによって認識されているオーケストレータのリストを返します。この API は、 `external_integration` 機能を備えた API キーで使用できます。

```
GET/openapi/v1/orchestrator/{scope}
```

パラメータ：（なし）

指定されたルート範囲のオーケストレータ オブジェクトのリストを返します。*scope* はルート範囲 ID にする必要があります。

オーケストレーションの作成

このエンドポイントは、新しいオーケストレーションを作成するために使用されます。

```
POST/openapi/v1/orchestrator/{scope}
```

vCenter オーケストレータのサンプル Python コード

```
req_payload = {
    "name": "VCenter Orchestrator"
    "type": "vcenter",
    "hosts_list": [ { "host_name": "8.8.8.8", "port_number": 443}],
    "username":"admin",
    "password":"admin"
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))
```

DNS オーケストレータのサンプル Python コード

```
req_payload = {
    "name": "DNS Server"
    "type": "dns",
    "hosts_list": [ { "host_name": "8.8.8.8", "port_number": 53}],
    "dns_zones": [ "lab.corp.com.", "dev.corp.com." ]
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))
```

Kubernetes オーケストレータのサンプル Python コード

```
req_payload = {
    "name": "k8s"
    "type": "kubernetes",
    "hosts_list": [ { "host_name": "8.8.8.8", "port_number": 53}],
    "certificate": "",
    "key": "",
    "ca_certificate": "",
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))
```

Ingress コントローラを使用した Kubernetes オーケストレータのサンプル Python コード

認証の詳細を作成するには、Kubernetes/OpenShift 外部オーケストレータに関する情報を参照してください。

```
req_payload = {
    "name": "k8s",
    "type": "kubernetes",
    "hosts_list": [ { "host_name": "8.8.8.8", "port_number": 53}],
    "certificate": "",
    "key": "",
    "ca_certificate": "",
    "ingress_controllers": [
        {
            "pod_selector": {
                "namespace": "ingress-nginx",
                "labels": [{ "key": "app", "value": "nginx-ingress"}],
            }
        }
    ]
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))
```

複数の Ingress コントローラを使用した Kubernetes オーケストレータのサンプル Python コード

認証の詳細を作成するには、Kubernetes/OpenShift 外部オーケストレータに関する情報を参照してください。

```
req_payload = {
    "name": "k8s",
    "type": "kubernetes",
    "hosts_list": [ { "host_name": "8.8.8.8", "port_number": 53}],
    "certificate": "",
    "key": "",
    "ca_certificate": "",
    "ingress_controllers": [
        {
            "pod_selector": {
                "namespace": "ingress-nginx",
                "labels": [{ "key": "app", "value": "nginx-ingress"}],
            },
            "controller_config": {
                "ingress_class": "nginx-class",
            }
        },
        {
            "pod_selector": {
                "namespace": "ingress-haproxy",
                "labels": [{ "key": "app", "value": "haproxy-ingress"}],
            },
            "controller_config": {
                "ingress_class": "haproxy-class",
                "http_ports": [8080],
                "https_ports": [8443],
                "namespace": "haproxy-watching-namespace"
            }
        }
    ],
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))

 ** Type AWS and EKS are no longer supported in external orchestrators. They have been
ported to
    connectors.
```

特定のオーケストレーションを取得する

このエンドポイントは、オーケストレーションのインスタンスを返します。

`GET/openapi/v1/orchestrator/{scope}/{orchestrator_id}`

指定された ID に関連付けられているオーケストレーション オブジェクトを返します。

オーケストレーションの更新

このエンドポイントはオーケストレーションを更新します。

`PUT/openapi/v1/orchestrator/{scope}/{orchestrator_id}`

パラメータ：

POST パラメータと同じ

特定のオーケストレーションの削除

このエンドポイントは、指定されたオーケストレーションを削除します。

`DELETE /openapi/v1/orchestrator/{scope}/{orchestrator_id}`

オーケストレータのゴールデンルール

この一連の API を使用して、外部 Kubernetes オーケストレータのゴールデンルールを管理できます。ゴールデンルールは、許可リスト強制モードで Kubernetes コントロールプレーンの接続を確保するために必要です。API キーに関連付けられた `external_integration` 機能が必要です。

ゴールデンルールで現在サポートされているオーケストレータタイプは「kubernetes」のみです。Kubernetes 以外のオーケストレータに対するこのエンドポイントへの要求は失敗します。

オーケストレータのゴールデン ルール オブジェクト

オーケストレーション オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
kubelet_port	integer	Kubelet ノードのローカル API ポート
サービス	配列	Kubernetes サービスオブジェクトの配列

オーケストレータのゴールデンルールを取得する

このエンドポイントは、関連付けられたゴールデンルールをオーケストレータに返します。この API は、external_integration 機能を備えた API キーで使用できます。

```
GET /openapi/v1/orchestrator/{scope}/{id}/gr
```

パラメータ：（なし）

単一のゴールデン ルール オブジェクトを返します。

ゴールデンルールの作成/更新

このエンドポイントは、既存のオーケストレータのゴールデンルールを作成または更新するために使用されます。

```
POST /openapi/v1/orchestrator/{scope}/{id}/gr
```

パラメータ：

属性	タイプ	説明
kubelet_port	integer	Kubelet ノードのローカル API ポート
サービス	配列	Kubernetes サービスオブジェクトの配列

サンプル python コード

```
  req_payload = {
              "kubelet_port":10255,
              "services": [
                          {
    "description": "kube-dns",
    "addresses":   [ "10.0.1.1:53/TCP", "10.0.1.1:53/UDP" ],
    "consumed_by":  [ "NODES", "PODS"],
   }
            ]
  }
  resp = restclient.post('/orchestrator/{scope_id}/{orchestrator_id}/gr',
json_body=json.dumps(req_payload))
```

FMC オーケストレータドメイン

この一連の API を使用して、外部 FMC オーケストレータのドメインを管理できます。特定の FMC ドメインでの適用を有効にするには、FMC ドメインが必要です。API キーに関連付けられた external_integration 機能が必要です。

現在サポートされている FMC ドメインのオーケストレータタイプは「fmc」のみです。non-fmc orches- trators でのこのエンドポイントへのリクエストは失敗します。

オーケストレータ FMC ドメインオブジェクト

オーケストレーション オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
fmc_domains	配列	FMC ドメインオブジェクトの配列

FMC ドメインオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
name	string	FMC ドメインの名前
enforcement_enabled	boolean	このフラグはデフォルトで false に設定されています。true の場合、外部オーケストレータは、ワークスペースに対してポリシーの適用が実行されるときに、「name」に一致するドメインにポリシーを展開します。

URL 属性については、以下で説明します。

属性	タイプ	説明
scope	string	インベントリが公開および表示されるテナントルート範囲の名前または ID
orchestrator_id	string	FMC オーケストレータのオーケストレータ ID

FMC ドメインの取得

このエンドポイントは、FMC オーケストレータに関連付けられている FMC で構成されている fmc ドメインを返します。この API は、 `external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/orchestrator/{scope}/{orchestrator_id}/fmcdomains
```

パラメータ：（なし）

FMC ドメインオブジェクトの属性のリストを含む json オブジェクトを返します。

FMC 外部オーケストレータの FMC ドメイン設定の更新

このエンドポイントは、既存の FMC 外部オーケストレータの FMC ドメイン属性を更新します。

```
PUT /openapi/v1/orchestrator/{scope}/{orchestrator_id}/fmcdomains
```

パラメータ：

属性	タイプ	説明
fmc_domains	配列	FMC ドメインオブジェクトの配列

FMC ドメインオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
name	string	FMC ドメインの名前
enforcement_enabled	boolean	このフラグはデフォルトで falseに設定されています。trueの場合、外部オーケストレータは、ワークスペースに対してポリシーの適用が実行されるときに、「name」に一致するドメインにポリシーを展開します。

URL 属性については、以下で説明します。

属性	タイプ	説明
scope	string	インベントリが公開および表示されるテナントルート範囲の名前または ID
orchestrator_id	string	FMC オーケストレータのオーケストレータ ID

サンプル python コード

```
req_payload = {
      "fmc_domains":    [
      {
          "enforcement_enabled": False,
          "name": "Global/Eng"
      },
      {
          "enforcement_enabled": True,
          "name": "Global/Prod"
      }
```

```
    ]
  }
  resp = restclient.put('/orchestrator/{scope}/{orchestrator_id}/fmcdomains',
json_body=json.dumps(req_payload))
```

RBAC（役割ベースのアクセス制御）に関する考慮事項

ルート範囲下でオーケストレータにアクセスするには、要求に使用されるAPIキーに要求の権限が必要です。すべてのオーケストレータ API 呼び出しは範囲が指定され、URL の一部として常にルート範囲 ID が必要です。オーケストレータは常にルート範囲レベルに存在し、サブ範囲の下に作成することはできません。特定のテナントルート範囲の下で作成されたオーケストレータ（およびこれらのオーケストレータによって学習されたインベントリ）は、他のテナントからは見えません。

複数のルートドメイン（vrfs）が構成されている可能性がある F5 ロードバランサの場合、F5 ルート ドメイン フィルタリング ロジックは、すべてのパーティションにわたって F5 上のすべてのエンティティをスキャンしますが、 F5 オーケストレータの *route_domain* フィールドで指定されたルートドメインとして評価されないエンティティ （サービス、snat プール、プール、およびバックエンド）は破棄します。

高可用性とフェールオーバーに関する考慮事項

hosts_listパラメータを使用すると、オーケストレータに複数のサーバーアドレスを設定できます。複数のサーバーアドレスの場合の Secure Workload サーバー選択ロジックは、オーケストレータタイプごとに異なります。

vCenter、Kubernetes、DNS、F5、NetScaler、Infoblox の場合、選択は最初の正常なエンドポイントに基づいて行われます。接続は永続的ではないため（Kubernetesを除く）、Secure Connector Orchestrator Managerは、ポーリング周期ごとにホストをスキャンし、hosts_listで検出された最初の正常なエンドポイントをポーリングします。Kubernetesの場合、永続的なイベントチャネルが維持され、接続が失敗すると、次の正常なエンドポイントを使用して、すべてのホストのスキャンと後続の完全なポーリングが実行されます。

Kubernetes RBAC リソースに関する考慮事項

Kubernetes クライアントは、次のリソースの GET/LIST/WATCH を試みます。

指定するKubernetes認証の資格情報には、次のリソースに対する最小限の権限セットが必要です。

リソース	**Verbs**
daemonsets	[get list watch]
deployments	[get list watch]

リソース	Verbs
endpoints	[get list watch]
名前空間	[get list watch]
ノード	[get list watch]
ポッド	[get list watch]
replicasets	[get list watch]
replicationcontrollers	[get list watch]
サービス	[get list watch]
statefulsets	[get list watch]
daemonsets.apps	[get list watch]
deployments.apps	[get list watch]
endpoints.apps	[get list watch]
namespaces.apps	[get list watch]
nodes.apps	[get list watch]
pods.apps	[get list watch]
replicasets.apps	[get list watch]
replicationcontrollers.apps	[get list watch]
services.apps	[get list watch]
statefulsets.apps	[get list watch]
daemonsets.extensions	[get list watch]
deployments.extensions	[get list watch]
endpoints.extensions	[get list watch]
namespaces.extensions	[get list watch]
nodes.extensions	[get list watch]
pods.extensions	[get list watch]
replicasets.extensions	[get list watch]
replicationcontrollers.extensions	[get list watch]
services.extensions	[get list watch]

リソース	Verbs
statefulsets.extensions	[get list watch]

Service Health

この API を使用して、Secure Workload クラスタで使用される全サービスの正常性をサービスの依存関係とともに取得できます。

（注） この API は、サイト管理者ユーザーのみが使用できます。

サービス正常性の取得

このエンドポイントは、サービス正常性情報を含む JSON オブジェクトを返します。

```
GET /openapi/v1/service_status
```

パラメータ：（なし）

応答オブジェクト： サービス正常性情報を含む JSON オブジェクト

サンプル python コード

```
  resp = restclient.get('/service_status')
```

Secure Connector

OpenAPI は、Secure Connector の機能を管理するエンドポイントを公開します。これらのエンドポイントでは、`external_integration` 機能が API キーに関連付けられている必要があります。

（注） Secure Connector API は、サイトレベルでは使用できません。これらはルート範囲レベルでのみ使用できます。

ステータスの取得

このエンドポイントは、指定されたルート範囲の Secure Connector トンネルの現在のステータスを返します。

```
GET /openapi/v1/secureconnector/name/{ROOT_SCOPE_NAME}/status
```

```
GET /openapi/v1/secureconnector/{ROOT_SCOPE_ID}/status
```

指定されたルート範囲に対する読み取りアクセス許可が必要です。

返されるステータスは、次のスキーマを持つ JSON オブジェクトです。

キー	タイプ	値
active	boolean	Secure Connector トンネルは現在アクティブです。
peer	string	トンネルの Secure Connector クライアント側の <ip>:<port>
start_time	int	トンネルが開始された時のタイムスタンプ（秒単位のエポック時間）
last_heartbeat	int	クライアントから最後に受信したハートビートのタイムスタンプ（秒単位のエポック時間）

トークンを取得

このエンドポイントは、指定されたルート範囲の Secure Connector クライアントをブートストラップするために使用される新しい 1 回限りの期間限定トークンを返します。

```
GET /openapi/v1/secureconnector/name/{ROOT_SCOPE_NAME}/token
```

```
GET /openapi/v1/secureconnector/{ROOT_SCOPE_ID}/token
```

指定されたルート範囲に対する OWNER 権限が必要です。

返されるトークンは、暗号で署名されたトークンを含む文字列で、1時間有効です。有効なトークンは、Secure Connector クライアントをブートストラップするために1回だけ使用できます。

証明書のローテーション

このエンドポイントは、指定されたルート範囲の新しい証明書の作成を強制的に実行します。新しい証明書は Secure Connector サーバーによって使用され、このルート範囲のクライアントからの証明書署名要求に署名するために使用されます。

```
POST /openapi/v1/secureconnector/name/{ROOT_SCOPE_NAME}/rotate_certs?invalidate_old=
‹→{true|false}
```

```
POST /openapi/v1/secureconnector/{ROOT_SCOPE_ID}/rotate_certs?invalidate_old=
‹→{true|false}
```

指定されたルート範囲に対する OWNER 権限が必要です。

このエンドポイントが呼び出されると、このルート範囲のクライアントとサーバー間の通信は、すぐに新しい証明書を使用するように移行します。

invalidate_old が false に設定されている場合、既存のクライアントは新しい公開キー/秘密キーのペアを自動的に作成し、既存の証明書を使用して、新しい公開キーの新しい証明書に署名します。

invalidate_old が true に設定されている場合、既存の証明書はすぐに無効になります。既存のクライアントはサーバーに接続できなくなり、新しいトークンを使用して再度ブートストラップする必要があります。詳細については、「Secure Connector の展開」を参照してください。

外部オーケストレータのポリシー適用ステータス

この一連の API は、*F5 BIG-IP* や *Citrix Netscaler* などのロードバランサ外部オーケストレータにポリシー適用ステータスを提供するために使用されます。



（注） これらの API を使用するには、VRF にアタッチされた範囲にユーザーがアクセスできる必要があります。

すべての外部オーケストレータのポリシー適用ステータスを取得する

このエンドポイントは、指定された VRF に属するすべての外部オーケストレータのポリシー適用ステータスを返します。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/tnp_policy_status/{vrfID}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
vrfID	integer	ルート範囲の VRF ID。

応答オブジェクト：ステータスが `ENFORCED` または `FAILED` または `IGNORED` のネットワークポリシーのリストを返します。

サンプル python コード

```
vrf_id = 676767
restclient.get('/tnp_policy_status/%d' % vrf_id)
```

外部オーケストレータのポリシー適用ステータスの取得

このエンドポイントは、指定された VRF に属する外部オーケストレータのポリシー適用ステータスを返します。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/tnp_policy_status/{vrfID}/{orchestratorID}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
vrfID	integer	ルート範囲の VRF ID。
orchestratorID	string	外部オーケストレータの ID。

応答オブジェクト：ステータスが ENFORCED または FAILED または IGNORED のネットワークポリシーのリストを返します。

サンプル python コード

```
vrf_id = 676767
orchestrator_id = '5ee3c991497d4f3b00f1ee07'
restclient.get('/tnp_policy_status/%d/%s' % (vrf_id, orchestrator_id))
```

管理対象データタップとデータシンクの証明書のダウンロード

この一連の API は、管理対象データタップとデータシンクの証明書をダウンロードするために使用されます。

（注） これらの API を使用するには、VRF にアタッチされた範囲にユーザーがアクセスできる必要があります。

特定の VRF ID に対するマネージド DataTap のリストの取得

このエンドポイントは、指定された VRF のマネージドデータタップのリストを返します。この API は、external_integration 機能を備えた API キーで使用できます。

```
GET /openapi/v1/mdt/{vrfID}
```

パラメータ：（なし）

マネージドデータタップ ID などの属性を持つマネージドデータタップのリストを返します。

指定された MDT ID に対するマネージド DataTap 証明書のダウンロード

このエンドポイントは、指定されたマネージドデータタップ ID の証明書をダウンロードするために使用されます。MDT ID は、上記のドキュメントで説明されているように、**/openapi/v1/mdt/{vrfID}** エンドポイントを使用して取得できます。この API は、external_integration 機能を備えた API キーで使用できます。

```
GET /openapi/v1/mdt/{vrfID}/{mdtID}/certs
```

パラメータ：

名前	タイプ	説明
format	string	キーストアと信頼ストアのフォーマット。値：`jks`（デフォルト値）または `cert`

次のファイルを含む tar.gz ファイルを返します。

`jks` 形式の場合：**truststore.jks**、**topic.txt**、**passphrase.txt**、**keystone.jks**、**kafkaBrokerIps.txt**、**consumer_name.txt**、**consumer_group_id.txt** .

`jks` 形式の場合：**KafkaConsumerCA.cert**、**KafkaConsumerPrivateKey.key**、**kafkaCA.cert**、**kafkaBrokerIps.txt**、**topic.txt**

KafkaConsumerCA.cert は公開証明書ファイルであり、**KafkaConsumerPrivateKey.key** ファイルには秘密キーがあります。**kafkaCA.cert** には CA 証明書があり、**kafkaBrokerIps.txt** には Kafka ブローカーの IP アドレスとポートのリストがあります。topic.txt ファイルには、MDT からデータを取得するために使用する必要があるトピックの名前が含まれています。**truststore.jks** および **keystone.jks** は、Java キーストアファイルです。

特定の VRF ID に対する DataSink のリストの取得

このエンドポイントは、指定された VRF の DataSink のリストを返します。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/datasinks/{vrfID}
```

パラメータ：（なし）

DataSink ID などの属性を持つ DataSink のリストを返します。

指定された DataSink ID の DataSink 証明書をダウンロードします。

このエンドポイントは、指定された DataSink ID の証明書をダウンロードするために使用されます。DataSink ID は、上記のドキュメントで説明されているように、**/openapi/v1/datasinks/{vrfID}** エンドポイントを使用して取得できます。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/datasinks/{vrfID}/{dsID}/certs
```

パラメータ：（なし）

userCA.cert、**userPrivateKey.key**、**intermediateCA.cert**、**kafkaCA.cert**、**kafkaBrokerIps.txt**、**topic.txt** ファイルを含む tar.gz ファイルを返します。

userCA.cert は公開証明書ファイルであり、**KafkaConsumerPrivateKey.key** ファイルには秘密キーがあります。**intermediateCA. cert** および **kafkaCA.cert** にはそれぞれ中間 CA 証明書とルート CA 証明書があります。**kafkaBrokerIps.txt** には Kafka ブローカーの IP アドレスとポートのリストがあります。**topic.txt** ファイルには、DataSink からデータを取得するために必要なトピックの名前があります。

変更ログ

この API は、ログ項目を変更するための読み取りアクセスを提供します。この API には、API キーに関連付けられた `user_role_scope_management` 機能が必要です。

（注） この API は、サイト管理者とルート範囲の所有者のみが使用できます。

変更ログオブジェクト

変更ログオブジェクトの属性の説明は以下のとおりです。

属性	タイプ	説明
id	string	変更ログ項目の固有識別子。
association_chain	オブジェクトの配列	この変更に関連する名前と ID のリスト。
scope	string	変更の範囲（Secure Workload の範囲とは異なります）。
action	string	アクションを変更します。
details	string	利用可能な場合、追加アクションの詳細。
created_At	integer	変更ログ項目が作成されたときの UNIX タイムスタンプ。
modifier	オブジェクト	変更を行うユーザー。
modified	オブジェクト	変更されたフィールドと値。
original	オブジェクト	変更前のフィールドと値。
version	integer	バージョン ID。

検索

このエンドポイントは、指定された条件に一致する変更ログのリストを返します。

```
GET /openapi/v1/change_logs
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_app_scope_id	string	（オプション）ルート範囲の所有者に必要です。ルート範囲で結果をフィルタリングします。
association_name	string	（オプション）ルート範囲の所有者に必要です。指定した項目タイプを返します。例：「H4Users」
history_action	string	（オプション）アクションを変更します。例：「update」
details	string	（オプション）アクションの詳細。例：「soft-delete」
before_epoch	integer	（オプション）この UNIX タイムスタンプより前に作成された結果を含めます。
after_epoch	integer	（オプション）この UNIX タイムスタンプの後に作成された結果を含めます。
offset	integer	（オプション）スキップする結果の数。
limit	integer	（オプション）結果の制限数。デフォルトは 1000 です。

応答オブジェクト：変更ログオブジェクトのリストを返します。

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

名前	タイプ	説明
total_count	integer	オフセットまたは制限を適用する前に一致した項目の総数。
Items	オブジェクトの配列	結果のリスト。

サンプル python コード

過去 1 日以内の指定されたルート範囲内の最後の 100 件の範囲オブジェクトの変更を取得します。

```
root_app_scope_id = '5ce480db497d4f1ca1fc2b2b'
one_day_ago = int(time.time() - 24*60*60)
resp = restclient.get('/change_logs', params={'root_app_scope_id': root_app_scope_id,
                                              'association_name': 'AppScope',
                                              'after_epoch': one_day_ago,
                                              'limit': 100})
```

2 番目の 1000 件の範囲オブジェクトの変更を取得します。

```
root_app_scope_id = '5ce480db497d4f1ca1fc2b2b'
resp = restclient.get('/change_logs', params={'root_app_scope_id': root_app_scope_id,
                                              'association_name': 'AppScope',
                                              'offset': 1000})
```

これらの結果をさらに絞り込み、新しい範囲の作成のみを表示します。

```
root_app_scope_id = '5ce480db497d4f1ca1fc2b2b'
one_day_ago = int(time.time() - 24 * 60 * 60)
resp = restclient.get('/change_logs', params={'root_app_scope_id': root_app_scope_id,
                                              'association_name': 'AppScope',
                                              'history_action': 'create',
                                              'after_epoch': one_day_ago,
                                              'limit': 100})
```

サイト管理者は、制限とオフセットを使用して、すべての範囲にわたるすべての変更を繰り返し取得できます。

```
resp = restclient.get('/change_logs', params={'offset': 100, 'limit': 100})
```

ルーティング不可能なエンドポイント

この一連の API は、ルーティング不可能なエンドポイントを管理したり、IP/サブネットがルーティング不可能であることを示したり、ユーザーによってルーティング不可能とマークされたエンドポイントのリストを取得したり、IP/サブネットに付いたルーティング不可能なエンドポイントのマークを解除したりするために使用されます。API キーに関連付けられた `user_data_upload` 機能が必要です。

ルーティング不可能なエンドポイントオブジェクト

ルーティング不可能なエンドポイントオブジェクトの属性について、以下で説明します。

属性	タイプ	説明
id	string	ルーティング不可能なエンドポイントの一意の識別子。
name	string	ルーティング不可能なエンドポイントのユーザー指定の名前。
subnet	string	IPv4/IPv6 サブネット。
vrf-id	long	ルーティング不可能なエンドポイントが属する VRF の ID。

属性	タイプ	説明
address_type	string	サブネットアドレスタイプに基づく IPv4/IPv6
host_uuid	string	エージェントの一意の ID
description	string	ルーティング不可能なエンドポイントのユーザー指定の説明。

ルーティング不可能なエンドポイントの取得

このエンドポイントは、指定されたテナント内のルーティング不可能なエンドポイントのリストを返します。

```
GET /openapi/v1/non_routable_endpoints/{rootScopeName}
```

パラメータ：（なし）

ルーティング不可能なエンドポイントの作成

このエンドポイントは、ルーティング不可能なエンドポイントを作成するために使用されます。

```
POST /openapi/v1/non_routable_endpoints/{rootScopeName}
```

パラメータ：

属性	タイプ	説明
name	string	ルーティング不可能なエンドポイントのユーザー指定の名前。
subnet	string	IPv4/IPv6 サブネット。
address_type（オプション）	string	サブネットアドレスタイプに基づく IPv4/IPv6
host_uuid（オプション）	string	エージェントの一意の ID
description（オプション）	string	ルーティング不可能なエンドポイントのユーザー指定の説明。

*オプションのフィールドが指定されていない場合、null 値が入力されます。

サンプル python コード

```
  req_payload = {
    "name": "nre-1",
    "subnet": "1.1.1.1/30",
    "address_type": IPV4,
    "description": "sample parameters test"
  }
  resp = restclient.post('/openapi/v1/non_routable_endpoints/Default',
json_body=json.dumps(req_payload))
```

名前を使用したルーティング不可能なエンドポイントの取得

このエンドポイントは、指定された名前のルーティング不可能なエンドポイントを返します。

```
GET /openapi/v1/non_routable_endpoints/{rootScopeName}/name/{name}
```

パラメータ：（なし）

ID を使用したルーティング不可能な特定のエンドポイントの取得

このエンドポイントは、指定された ID のルーティング不可能なエンドポイントを返します。

```
GET /openapi/v1/non_routable_endpoints/{rootScopeName}/id/{id}
```

パラメータ：（なし）

ルーティング不可能な特定のエンドポイント名の更新

このエンドポイントは、ルーティング不可能なエンドポイントを更新するために使用されます。既存のルーティング不可能なエンドポイントの ID または名前を使用して、エンドポイント名前を更新します。

```
PUT /openapi/v1/non_routable_endpoints/{rootScopeName}
```

パラメータ：

属性	タイプ	説明
id	string	ルーティング不可能なエンドポイントの一意の識別子。
name	string	ルーティング不可能なエンドポイントのユーザー指定の名前。
new_name	string	更新する新しい名前

サンプル python コード

```
  req_payload = {
    "name": "nre-1",
    "new_name": "nre-updated",
  }
```

```
  resp = restclient.put('/openapi/v1/non_routable_endpoints/Default',
json_body=json.dumps(req_payload))


  req_payload = {
    "id": "5f706964a5b5f16ed4b0aacb",
    "new_name": "nre-updated",
  }
  resp = restclient.put('/openapi/v1/non_routable_endpoints/Default',
json_body=json.dumps(req_payload))
```

名前を使用したルーティング不可能なエンドポイントの削除

このエンドポイントは、特定のルーティング不可能なエンドポイントを削除します。

```
DELETE /openapi/v1/non_routable_endpoints/{rootScopeName}/name/{name}
```

名前を使用したルーティング不可能なエンドポイントの削除

このエンドポイントは、特定のルーティング不可能なエンドポイントを削除します。

```
DELETE /openapi/v1/non_routable_endpoints/{rootScopeName}/id/{id}
```

外部アプライアンスおよびコネクタの設定およびコマンドのスキーマ

設定グループ API

設定グループ API は、アプライアンスおよびコネクタ API の設定スキーマを提供します。これらの API のセットには、API キーに関連付けられている `sensor_management` または `external_integration` 機能が必要です。

設定のスキーマを取得する API

このエンドポイントは、選択された設定のタイプやグループの静的設定スキーマを返します。

```
GET /openapi/v1/config_groups/schema/<type>
```

ここで、<type> はアプライアンス設定タイプです。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
タイプ	string	[redacted] から設定タイプを指定します

応答オブジェクト：選択された設定タイプの設定スキーマを返します。

サンプル応答

```
resp = restclient.get('/config_groups/schema/LOG')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答


```
{
    "type": "LOG",
    "name": "Log",
    "mode": "TEST",
    "config": {
      "secured": {},
      "unsecured": {
        "log-level": "info",
        "max-log-size": 10,
        "max-log-age": 30,
        "max-log-backups": 20 }
    },
    "fill_ins": [
      {
        "field": "log-level",
        "label": "Logging Level",
        "placeholder": "info",
        "type": "user_fill_in",
        "input_type": "dropdown",
        "possible_values": [
          "trace",
          "debug",
          "info",
          "warn",
          "error"
        ]
    },
      {
        "field": "max-log-size",
        "label": "Max Log File Size (in MB)",
        "placeholder": 10,
        "type": "user_fill_in",
        "input_type": "number",
        "min": 1,
        "max": 10240
      },
      {"field": "max-log-age",
        "label": "Log Rotation (in days)",
        "placeholder": 30,
        "type": "user_fill_in",
        "input_type": "number",
        "min": 1,
        "max": 365
      },
      {
        "field": "max-log-backups",
        "label": "Log Rotation (in instances)",
        "placeholder": 20,
        "type": "user_fill_in",
        "input_type": "number",
        "min": 1,
        "max": 100
}
    ]
  }
```

トラブルシューティング コマンドのスキーマを取得する API

このエンドポイントは、選択したタイプのトラブルシューティング コマンドの、静的なトラブルシューティング コマンドの設定スキーマを返します。

```
GET /openapi/v1/config_groups/command_schema/<type>
```

リクエストペイロードで、<type> はトラブルシューティング コマンドの設定タイプです。

パラメータ：

リクエスト URL には次のパラメータが含まれています

名前	タイプ	説明
タイプ	string	"SHOW [redacted] からコマンドタイプを指定します

応答オブジェクト：選択された設定タイプの設定スキーマを返します。

サンプル応答

```
resp = restclient.get('/config_groups/command_schema/SHOW_LOG')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "name": "Show logs",
    "desc": "Show the contents of a log file",
    "long_desc": "Show the contents of a log file and optionally grep the file for a
specified pattern. The output is tailed for the last 5000 lines.",
    "valid_appliances": [
      "TETRATION_DATA_INGEST",
      "TETRATION_EDGE",
      "TETRATION_EXPORT"
    ],
    "valid_connectors": [
      "netflow",
      "netscaler",
      "f5",
      "aws",
      "anyconnect",
      "slack",
      "kinesis",
      "syslog",
      "email",
      "pagerduty",
      "ise",
      "asa",
      "meraki",
      "servicenow",
      "wad"
    ],
    "arg_fillins": [
      {
        "field": "pattern",
        "label": "Grep Pattern",
        "input_type": "text",
```

```
        "optional": true
      }
    ],
    "output_type": "FILE",
    "output_ext": "LOG"
  }
```

外部アプライアンス

外部アプライアンス API

外部アプライアンス API は、Cisco Secure Workload 外部アプライアンスの管理に関連付けられています。これらの API のセットには、API キーに関連付けられている `sensor_management` または `external_integration` 機能が必要です。

アプライアンスのリストを取得する API

このエンドポイントは、アプライアンスのリストを返します。

```
GET /openapi/v1/ext_appliances?root_scope_id=<root_scope_id>&type=<type>
```

ここで、<root_scope_id> は「範囲の取得」の API から取得できる root_scope_id です。<type> はアプライアンスタイプを決定する文字列です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲を指定します
type	string	アプライアンスタイプを指定します。“TETRATION_EDGE”、“TETRATION_DATA_INGEST”、“TETRATION_EXPORT”、“TETRATION_ERSPAN”、または“TETRATION_INTERNAL” の値を使用できます

応答オブジェクト：アプライアンスのリストを返します。

サンプル応答

```
resp =
restclient.get('/ext_appliances?root_scope_id=63bf8d2f497d4f7287dbd335&type=TETRATION_INTERNAL')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
[
    {
```

```
    "root_scope_id": "63bf8d2f497d4f7287dbd335",
    "type": "tetration_internal",
    "status": {
      "state": "active",
      "controller_state": "up",
      "message": "",
      "display_state": "active"
    },
    "auto_upgrade": true,
    "created_at": 1673498141,
    "updated_at": 1673498141,
    "registered_at": 1673498141,
    "last_checkin_at": 0,
    "last_rpm_sent_at": 0,
    "upgrade_attempts": 0,
    "delete_attempts": 0,
    "last_delete_msg_sent_at": 0,
    "taas": false,
    "deleted": false,
    "deleted_at": 0,
    "connector_limit": 5000,
    "available_slots": 5000,
    "internal": true,
    "id": "63bf8e1d6419d06bef39bc85",
    "ha_peer_appliance_id": "",
    "display_type": "Tetration Internal"
  }
]
```


アプライアンスを作成する API

このエンドポイントは、アプライアンスを作成します。

```
POST /openapi/v1/ext_appliances
```

リクエストペイロードで、<config> スキーマを取得するには、アプライアンススキーマを取得する API（1261 ページ）の応答からいずれかの “valid_config” を選択し、設定のスキーマを取得する API（1248 ページ）に “valid_config” を適用します。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
name	string	名前を指定します
root_scope_id	string	ルート範囲を指定します
type	string	アプライアンスタイプを指定します。“TETRATION_EDGE”、“TETRATION_DATA_INGEST”、“TETRATION_EXPORT”、“TETRATION_ERSPAN”、または “TETRATION_INTERNAL” の値を使用できます

名前	タイプ	説明
config	set	入力された設定スキーマを JSON 形式で提供します
taas	boolean	アプライアンスが TAAS 環境用かどうかを示します
version	string	バージョンを指定します

応答オブジェクト：作成されたアプライアンスを返します。

サンプル応答

```
req_payload = {
    "name":"Data Ingest Appliance",
    "type":"tetration_data_ingest",
    "root_scope_id":"63c41ff2497d4f5f5be73662",
    "config":{
      "VM3":{
        "secured":{},
        "unsecured":{
          "cidr":[
            "172.26.231.141/23",
            "172.26.231.142/23",
            "172.26.231.143/23"
          ],
          "gateway":[
            "172.26.231.140",
            "172.26.231.140",
            "172.26.231.140"
          ],
          "cidr_v6":[],
          "gateway_v6":[],
          "dns":[
            "testserver"
          ],
          "search_domains":[],
          "hostname":"",
          "use_proxy_for_tetration":false,
          "https_proxy":"",
          "no_proxy":[],
          "docker_subnet":""
        }
      }
    }
  }
  resp = restclient.post('/ext_appliances', json_body=json.dumps(req_payload))
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
 {
    "root_scope_id":"63c41ff2497d4f5f5be73662",
    "type":"tetration_data_ingest",
    "status":{
      "state":"pending_dio",
      "controller_state":"down",
      "message":"Setting up appliance",
      "display_state":"preparing"
```

```
  },
  "auto_upgrade":true,
  "created_at":1674183549,
  "updated_at":1674183549,
  "registered_at":0,
  "last_checkin_at":0,
  "last_rpm_sent_at":0,
  "upgrade_attempts":0,
  "delete_attempts":0,
  "last_delete_msg_sent_at":0,
  "name":"Data Ingest Appliance",
  "taas":false,
  "deleted":false,
  "deleted_at":0,
  "connector_limit":3,
  "available_slots":0,
  "internal":false,
  "id":"63ca037dbca44e263daeb5d0",
  "ha_peer_appliance_id":"",
  "display_type":"Tetration Data Ingest"
}
```


アプライアンスを削除する API

このエンドポイントは、指定されたアプライアンスを削除します。

```
DELETE /openapi/v1/ext_appliances/<id>
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します

応答オブジェクト：削除されたアプライアンスのステータスを返します。

サンプル応答

```
resp = restclient.delete('/ext_appliances/63be3b1ade36423c12bff6e1')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答


```
 {
    "status": 200,
    "code": 1000,
    "message": "deleted"
  }
```


ID でアプライアンスを取得する API

このエンドポイントは、指定された ID を持つアプライアンスを取得します。

```
GET /openapi/v1/ext_appliances/<id>
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します

応答オブジェクト：指定された ID を持つアプライアンスを返します。

サンプル応答

```
resp = restclient.get('/ext_appliances/63bf8e1d6419d06bef39bc85')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答


```
{
    "root_scope_id": "63bf8d2f497d4f7287dbd335",
    "type": "tetration_internal",
    "status": {
      "state": "active",
      "controller_state": "up",
      "message": "",
      "display_state": "active"
    },
    "auto_upgrade": true,
    "created_at": 1673498141,
    "updated_at": 1673498141,
    "registered_at": 1673498141,
    "last_checkin_at": 0,
    "last_rpm_sent_at": 0,
    "upgrade_attempts": 0,
    "delete_attempts": 0,
    "last_delete_msg_sent_at": 0,
    "taas": false,
    "deleted": false,
    "deleted_at": 0,
    "connector_limit": 5000,
    "available_slots": 5000,
    "internal": true,
    "id": "63bf8e1d6419d06bef39bc85",
    "ha_peer_appliance_id": "",
    "display_type": "Tetration Internal"
  }
```


アプライアンス名を変更する API

このエンドポイントは、指定された ID を持つアプライアンスの名前を変更します。

```
PUT /openapi/v1/ext_appliances/<id>
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します
name	string	アプライアンスの新しい名前を指定します

応答オブジェクト：指定された ID を持つアプライアンスと新しい名前を返します。

サンプル応答

```
req_payload = {
    "name": "new_name",
  }
  resp = restclient.put('/ext_appliances/63bf8e1d6419d06bef39bc85',
json_body=json.dumps(req_payload))
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "root_scope_id": "63bf8d2f497d4f7287dbd335",
    "type": "tetration_internal",
    "status": {
      "state": "active",
      "controller_state": "up",
      "message": "",
      "display_state": "active"
    },
    "auto_upgrade": true,
    "created_at": 1673498141,
    "updated_at": 1673498141,
    "registered_at": 1673498141,
    "last_checkin_at": 0,
    "last_rpm_sent_at": 0,
    "upgrade_attempts": 0,
     "delete_attempts": 0,
    "last_delete_msg_sent_at": 0,
    "name": "new_name",
    "taas": false,
    "deleted": false,
    "deleted_at": 0,
    "connector_limit": 5000,
    "available_slots": 5000,
    "internal": true,
    "id": "63bf8e1d6419d06bef39bc85",
    "ha_peer_appliance_id": "",
    "display_type": "Tetration Internal"
  }
```

設定タイプでの設定を取得する API

このエンドポイントは、指定された ID と設定タイプを持つアプライアンスの設定を取得します。

```
GET /openapi/v1/ext_appliances/<id>/config?config_type=<config_type>
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。<config_type> はアプライアンススキーマを取得する API （1261 ページ）から取得できる "valid_config" です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します
config_type	string	設定タイプを指定します。使用できるすべての値については、設定のスキーマを取得する API （1248 ページ）の説明にリストされている値を参照してください

応答オブジェクト：指定されたアプライアンス ID と設定タイプを持つ設定を返します。

サンプル応答

```
resp =
restclient.get('/ext_appliances/63c1272039042a1c0ddd3e20/config?config_type=VM3_ERSPAN')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
[
    {
      "mode": "TEST_AND_APPLY",
      "name": "VM3_ERSPAN",
      "root_scope_id": "63bf8d2f497d4f7287dbd335",
      "vrf_id": 1,
      "appliance_id": "63c1272039042a1c0ddd3e20",
      "deleted": false,
      "type": "VM3_ERSPAN",
      "state": "COMMIT",
      "attempts": 0,
      "config": {
        "secured": {},
        "unsecured": {
          "cidr": [
            "172.26.231.141/23",
            "172.26.231.142/23",
            "172.26.231.143/23"
          ],
          "gateway": [
            "172.26.231.140",
            "172.26.231.140",
            "172.26.231.140"
          ],
          "cidr_v6": [],
          "gateway_v6": [],
          "dns": [
            "test"
          ],
```

```
            "search_domains": [],
            "hostname": "hjtest",
            "https_proxy": "",
            "no_proxy": []
          }
        },
        "push_to_dio_at": 0,
        "created_at": 1673602848,
        "updated_at": 1673602848,
        "discovery_completed_at": 0,
        "committed_at": 0,
        "delete_at": 0,
        "error_at": 0,
        "hidden": false,
        "discovery_phase": 0,
        "internal": false,
        "id": "63c1272039042a1c0ddd3e21"
      }
    ]
```

新しい設定を外部アプライアンスに追加する API

このエンドポイントは、指定された ID を持つアプライアンスに新しい設定を追加します

```
POST /openapi/v1/ext_appliances/<id>/config
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。リクエストペイロードでは、<type> はアプライアンススキーマを取得する API （1261 ページ）から取得できる "valid_config" です。<config> スキーマを取得するには、アプライアンススキーマを取得する API （1261 ページ）の応答からいずれかの "valid_config" を選択し、設定のスキーマを取得する API （1248 ページ）に "valid_config" を適用します。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
name	string	設定名を指定します
type	string	設定タイプを指定します。使用できるすべての値については、設定のスキーマを取得する API （1248 ページ）の説明にリストされている値を参照してください
config	set	入力された設定スキーマを JSON 形式で提供します

応答オブジェクト：更新された設定を返します。

サンプル応答

```
req_payload = {
    "name": "new_config",
    "type": "VM3_ERSPAN",
    "config": {}
  }
  resp = restclient.post('/ext_appliances/63c1272039042a1c0ddd3e20/config',
```

```
json_body=json.dumps(req_payload))
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答


```
{
   "prev_id": "63c1272039042a1c0ddd3e21",
   "mode": "TEST_AND_APPLY",
   "name": "new_config",
   "root_scope_id": "63bf8d2f497d4f7287dbd335",
   "vrf_id": 1,
   "appliance_id": "63c1272039042a1c0ddd3e20",
   "deleted": false,
   "type": "VM3_ERSPAN",
   "state": "COMMIT",
   "attempts": 0,
   "config": {
     "secured": {},
     "unsecured": null
   },
   "push_to_dio_at": 0,
   "created_at": 1673661042,
   "updated_at": 1673661042,
   "discovery_completed_at": 0,
   "committed_at": 0,
   "delete_at": 0,
   "error_at": 0,
   "hidden": false,
   "discovery_phase": 0,
   "internal": false,
   "id": "63c20a7239042a0991b871b7"
 }
```


設定を削除するための API

このエンドポイントは、指定されたアプライアンスから設定を削除します。

```
DELETE /openapi/v1/ext_appliances/<id>/config/<config_id>
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。<config_id> は設定タイプでの設定を取得する API（1256 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します
config_id	string	設定 ID を指定します

応答オブジェクト：指定されたアプライアンスの削除された設定のステータスを返します。

サンプル応答

```
resp =
restclient.delete('/ext_appliances/63c1272039042a1c0ddd3e20/config/63c1272039042a1c0ddd3e21')
```

```
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "status": 200,
    "code": 1000,
    "message": "deleted"
  }
```

設定を取得する API

このエンドポイントは、指定された ID を持つ設定を取得します。

```
GET /openapi/v1/ext_appliances/<id>/config/<config_id>
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。<config_id> は設定タイプでの設定を取得する API （1256 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します
config_id	string	設定 ID を指定します

応答オブジェクト：指定された ID を持つ設定を返します。

サンプル応答

```
resp =
restclient.get('/ext_appliances/63c1272039042a1c0ddd3e20/config/63c1272039042a1c0ddd3e21')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
b
```

サンプル応答

```
{
    "mode": "TEST_AND_APPLY",
    "name": "VM3_ERSPAN",
    "root_scope_id": "63bf8d2f497d4f7287dbd335",
    "vrf_id": 1,
    "appliance_id": "63c1272039042a1c0ddd3e20",
    "deleted": false,
    "type": "VM3_ERSPAN",
    "state": "COMMIT",
    "attempts": 0,
    "config": {
      "secured": {},
      "unsecured": {
        "cidr": [
          "172.26.231.141/23",
          "172.26.231.142/23",
          "172.26.231.143/23"
```

```
        ],
        "gateway": [
          "172.26.231.140",
          "172.26.231.140",
          "172.26.231.140"
        ],
        "cidr_v6": [],
        "gateway_v6": [],
        "dns": [
          "test"
        ],
        "search_domains": [],
        "hostname": "hjtest",
        "https_proxy": "",
        "no_proxy": []
      }
    },
    "push_to_dio_at": 0,
    "created_at": 1673602848,
    "updated_at": 1673602848,
    "discovery_completed_at": 0,
    "committed_at": 0,
    "delete_at": 0,
    "error_at": 0,
    "hidden": false,
    "discovery_phase": 0,
    "internal": false,
    "id": "63c1272039042a1c0ddd3e21"
  }
```


アプライアンススキーマを取得する API

このエンドポイントは、指定されたタイプを持つアプライアンススキーマを取得します。

```
GET /openapi/v1/ext_appliances/schema/<type>
```

ここで、<type> はアプライアンスタイプを決定する文字列です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
タイプ	string	アプライアンスタイプを指定します。"TETRATION_ERSPAN"、"TETRATION_F5"、"TETRATION_NETSCALER"、または "TETRATION_INTERNAL" の値を使用できます

応答オブジェクト：設定スキーマを返します。

サンプル応答

```
resp = restclient.get('/ext_appliances/schema/TETRATION_ERSPAN')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
 {
    "name": "Data Ingest for ERSPAN",
    "type": "tetration_erspan",
    "desc": "Data Ingest Appliance for ERSPAN",
    "long_desc": "Data Ingest appliance for ERSPAN is a software appliance that can
export flow data from ERSPAN appliance.",
    "valid_config": [
      "VM3_ERSPAN"
    ],
    "deploy_config": [
      "VM3_ERSPAN"
    ],
    "connectors": [
      "ERSPAN"
    ],
    "limit_connectors_per_appliance": 0,
    "limit_per_rootscope": 8,
    "limit_per_rootscope_taas": 4,
    "limit_per_cluster": 150,
    "cco_url":
"https://software.cisco.com/download/home/286309796/type/286309874/release/3.7.1.26.devel"

  }
```

-

このアプライアンスで使用可能なすべてのトラブルシューティング コマンドを表示する API

このエンドポイントは、指定されたアプライアンスで使用できるトラブルシューティング コマンドのリストを返します。

```
GET /openapi/v1/ext_appliances/<id>/commands/available
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します

応答オブジェクト：指定されたアプライアンスで使用できるトラブルシューティング コマンドのリストを返します。

サンプル応答

```
resp = restclient.get('/ext_appliances/63c6ef42bca44e2b5e729191/commands/available')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
[
    {
      "type": "UPDATE_LISTENING_PORT",
      "name": "Update the listening port on a connector"
    },
    {
      "type": "SNAPSHOT_APPLIANCE",
```

```
    "name": "Collect Snapshot from Appliance"
  },
  {
    "type": "LIST_FILES",
    "name": "List a directory"
  },
  {
    "type": "CONTROLLER_PROFILING",
    "name": "Collect controller profile"
  },
  {
    "type": "SHOW_LOG",
    "name": "Show logs"
  },
  {
    "type": "SHOW_SUPERVISOR_COMMANDS",
    "name": "Execute supervisorctl command"
  },
  {
    "type": "PACKET_CAPTURE",
    "name": "Packet capture"
  },
  {
    "type": "NETWORK_CONNECTIVITY_COMMANDS",
    "name": "Test network connectivity"
  },
  {
    "type": "SHOW_DOCKER_COMMANDS",
    "name": "Execute docker command"
  },
  {
    "type": "CONNECTOR_ALERT_INTERVAL_APPLIANCE",
    "name": "Override connector alert interval for Appliance"
  },
  {
    "type": "SHOW_RUNNING_CONF",
    "name": "Show running configuration"
  },
  {
    "type": "SHOW_SYS_COMMANDS",
    "name": "Execute system command"
  }
]
```


すべてのトラブルシューティング コマンドを表示する API

このエンドポイントは、指定されたアプライアンスに対して有効になっているトラブルシューティング コマンドのリストを返します。

```
GET /openapi/v1/ext_appliances/<id>/commands
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します

応答オブジェクト：指定されたアプライアンスに対して有効になっているトラブルシューティング コマンドのリストを返します。

サンプル応答

```
resp = restclient.get('/ext_appliances/63be3b1ade36423c12bff6e1/commands')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
[
    {
      "appliance_id": "63be3b1ade36423c12bff6e1",
      "state": "pending",
      "level": "APPLIANCE",
      "command": "SHOW_LOG",
      "arg_string": "",
      "args": {},
      "tailed": false,
      "rc": 0,
      "push_to_dio_at": 0,
      "attempts": 0,
      "deleted": false,
      "deleted_at": 0,
      "created_at": 1673595392,
      "total_chunk": 0,
      "id": "63c10a0039042a6aee1b008c"
    }
  ]
```

トラブルシューティング コマンドを作成する API

このエンドポイントは、指定されたアプライアンスで使用できるトラブルシューティング コマンドを作成します。

```
POST /openapi/v1/ext_appliances/<id>/commands
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。リクエストペイロードで、<command> はトラブルシューティング コマンドのスキーマを取得する API （1250 ページ）の応答の "valid_appliances" フィールドから取得できるトラブルシューティング コマンド タイプです。<arguments> はコマンドスキーマの入力された JSON オブジェクトであり、トラブルシューティング コマンドのスキーマを取得する API （1250 ページ）から取得できます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します
command	string	コマンドタイプを指定します
引数	set	入力されたコマンドスキーマを JSON 形式で提供します

応答オブジェクト：指定されたアプライアンスの作成されたトラブルシューティング コマンドを返します。

サンプル応答

```
req_payload = {
    "command": "SHOW_LOG",
    "arguments": {}
  }
  resp = restclient.post('/ext_appliances/63be3b1ade36423c12bff6e1/commands',
json_body=json.dumps(req_payload))
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
 {
    "appliance_id": "63be3b1ade36423c12bff6e1",
    "state": "pending",
    "level": "APPLIANCE",
    "command": "SHOW_LOG",
    "args": {},
    "tailed": false,
    "rc": 0,
    "push_to_dio_at": 0,
    "attempts": 0,
    "deleted": false,
    "deleted_at": 0,
    "created_at": 1673595392,
    "total_chunk": 0,
    "id": "63c10a0039042a6aee1b008c"
  }
```

-

トラブルシューティング コマンドを削除する API

このエンドポイントは、指定されたアプライアンスに対して有効になっているトラブルシューティング コマンドを削除します。

```
DELETE /openapi/v1/ext_appliances/<id>/commands/<command_id>
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。<command_id> はすべてのトラブルシューティング コマンドを表示する API （1263 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します
command_id	string	コマンド ID を指定します

応答オブジェクト：指定されたアプライアンスの削除されたトラブルシューティング コマンドのステータスを返します。

サンプル応答

```
resp =
restclient.delete('/ext_appliances/63be3b1ade36423c12bff6e1/commands/63c10a0039042a6aee1b008c')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
   "status": 200,
   "code": 1000,
   "message": "deleted"
 }
```

トラブルシューティング コマンドをルックアップする API

このエンドポイントは、指定されたアプライアンスに対して選択されたトラブルシューティング コマンドを返します。

```
GET /openapi/v1/ext_appliances/<id>/commands/<command_id>
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。<command_id> はすべてのトラブルシューティング コマンドを表示する API （1263 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します
command_id	string	コマンド ID を指定します

応答オブジェクト：指定されたアプライアンスに対して選択されたトラブルシューティング コマンドを返します。

サンプル応答

```
resp =
restclient.get('/ext_appliances/63be3b1ade36423c12bff6e1/commands/63c10fd139042a1c0ddd3e1f')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "appliance_id": "63be3b1ade36423c12bff6e1",
    "state": "pending",
    "level": "APPLIANCE",
    "command": "SHOW_LOG",
    "arg_string": "",
    "args": {},
    "tailed": false,
    "rc": 0,
    "push_to_dio_at": 0,
    "attempts": 0,
    "deleted": false,
```

```
    "deleted_at": 0,
    "created_at": 1673596881,
    "total_chunk": 0,
    "id": "63c10fd139042a1c0ddd3e1f"
}
```


アプライアンスコマンドの出力をファイルとしてダウンロードする API

このエンドポイントは、コマンドの出力をファイルとしてダウンロードします。

```
GET /openapi/v1/appliances/<id>/commands/{command_id}/download
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。<command_id> はすべてのトラブルシューティング コマンドを表示する API （1263 ページ）から取得できる id です。すべてのコマンドにダウンロード可能な出力があるわけではありません。トラブルシューティング コマンドのスキーマを取得する API （1250 ページ）によって提供されるコマンドスキーマを確認してください。ここで、"output_type": "FILE" はダウンロード可能なコンテンツがあることを示し、"output_ext" はファイルタイプを示しています。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します
command_id	string	コマンド ID を指定します

応答オブジェクト：コマンド出力をファイルとしてダウンロードします。

サンプル応答

```
resp = restclient.download('downloadFile',
'/appliances/63c6ef42bca44e2b5e729191/commands/63cace941a49bd4c0e0cf45a/download')
```

コネクタ

コネクタ API

コネクタ API は、Cisco Secure Workload コネクタの管理に関連付けられています。これらの API のセットには、API キーに関連付けられている `sensor_management` または `external_integration` 機能が必要です。

すべてのタイプのコネクタを取得する API

このエンドポイントは、指定されたルート範囲のすべてのタイプのコネクタを取得します。

```
GET /openapi/v1/connectors/cards?root_scope_id=<root_scope_id>
```

ここで、<root_scope_id> は範囲の取得 （1122 ページ）の API から取得できる root_scope_id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲を指定します

応答オブジェクト：指定されたルート範囲 ID を持つすべてのタイプのコネクタを返します。

サンプル応答

```
resp = restclient.get('/connectors/cards?root_scope_id=63bf8d2f497d4f7287dbd335')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
[{
    "type": "NETFLOW",
    "name": "NetFlow",
    "desc": "Collect telemetry from network devices",
    "long_desc": "Collect NetFlow V9 and/or IPFIX telemetry from network devices such
as routers and switches.",
    "group": "Flow Ingest",
    "index": 0,
    "appliance_type": "tetration_data_ingest",
    "state": "disabled",
    "limit_per_appliance": 3,
    "limit_per_rootscope": 10,
    "limit_per_cluster": 100,
    "config": [
      "LOG",
      "ALERT"
    ],
    "max_instances": 0,
    "noteworthy": false,
    "hidden": false,
    "capabilities": [
      "Flow Visibility"
    ],
    "connector_count": 0,
    "group_order": 0
  },
  {
    "type": "NETSCALER",
    "name": "NetScaler",
    "desc": "Collect telemetry from Citrix ADC",
    "long_desc": "Collect telemetry from Citrix ADC, stitch client and server side
flows.",
    "group": "Flow Ingest",
    "index": 2,
    "appliance_type": "tetration_data_ingest",
    "state": "disabled",
    "limit_per_appliance": 3,
    "limit_per_rootscope": 10,
    "limit_per_cluster": 100,
    "config": [
      "LOG",
      "ALERT"
    ],
    "max_instances": 0,
    "noteworthy": false,
    "hidden": false,
    "capabilities": [
      "Flow Visibility",
```

```
      "Flow Stitching",
      "ADM"
    ],
    "connector_count": 0,
    "group_order": 0
  },
  {
    "type": "F5",
    "name": "F5",
    "desc": "Collect telemetry from F5 BIG-IP",
    "long_desc": "Collect telemetry from F5 BIG-IP, stitch client and server side flows,
enrich client inventory with user attributes.",
    "group": "Flow Ingest",
    "index": 1,
    "appliance_type": "tetration_data_ingest",
    "state": "disabled",
    "limit_per_appliance": 3,
    "limit_per_rootscope": 10,
    "limit_per_cluster": 100,
    "config": [
      "LDAP",
      "LOG",
      "ALERT"
    ],
    "max_instances": 0,
    "noteworthy": false,
    "hidden": false,
    "capabilities": [
      "Flow Visibility",
      "User Insights",
      "Flow Stitching",
      "ADM"
    ],
    "connector_count": 0,
    "group_order": 0
  },
  {
    "type": "ANYCONNECT",
    "name": "AnyConnect",
    "desc": "Collect telemetry from AnyConnect NVM",
    "long_desc": "Collect telemetry data from Cisco AnyConnect Network Visibility Module
(NVM) and enrich endpoint inventories with user attributes",
    "group": "Endpoints",
    "index": 0,
    "appliance_type": "tetration_data_ingest",
    "state": "disabled",
    "limit_per_appliance": 1,
    "limit_per_rootscope": 50,
    "limit_per_cluster": 500,
    "config": [
      "ENDPOINT",
      "LDAP",
      "LOG",
      "ALERT"
    ],
    "max_instances": 0,
    "noteworthy": false,
    "hidden": false,
    "capabilities": [
      "Flow Visibility",
      "Process Annotation",
      "User Insights",
      "Endpoint Insights",
      "Inventory Enrichment"
```

```
    ],
    "connector_count": 0,
    "group_order": 1
  },
  {
    "type": "ASA",
    "name": "Cisco Secure Firewall",
    "desc": "Collect telemetry from Cisco Secure Firewall",
    "long_desc": "Collect telemetry from Cisco Secure Firewall, stitch client and server
side flows. Recommended usage with ISE connector for flow visibility.",
    "group": "Flow Ingest",
    "index": 3,
    "appliance_type": "tetration_data_ingest",
    "state": "disabled",
    "limit_per_appliance": 1,
    "limit_per_rootscope": 10,
    "limit_per_cluster": 100,
    "config": [
      "LOG",
      "ALERT"
    ],
    "max_instances": 0,
    "noteworthy": false,
    "hidden": false,
    "capabilities": [
      "Flow Visibility",
      "Flow Stitching",
      "ADM"
    ],
    "connector_count": 0,
    "group_order": 0
  },
  {
    "type": "SLACK",
    "name": "Slack",
    "desc": "Send alerts to Slack",
    "long_desc": "Send Tetration Alerts to Slack.",
    "group": "Alerts",
    "index": 2,
    "appliance_type": "tetration_edge",
    "state": "disabled",
    "limit_per_appliance": 1,
    "limit_per_rootscope": 1,
    "limit_per_cluster": 150,
    "config": [
      "SLACK_NOTIFIER",
      "ALERT"
    ],
    "max_instances": 0,
    "noteworthy": false,
    "hidden": false,
    "capabilities": [
      "Alert Destination"
    ],
    "connector_count": 0,
    "group_order": 3
  },
  {
    "type": "KINESIS",
    "name": "Kinesis",
    "desc": "Send alerts to Kinesis",
    "long_desc": "Send Tetration Alerts to Kinesis.",
    "group": "Alerts",
    "index": 4,
```

```
  "appliance_type": "tetration_edge",
  "state": "disabled",
  "limit_per_appliance": 1,
  "limit_per_rootscope": 1,
  "limit_per_cluster": 150,
  "config": [
    "KINESIS_NOTIFIER",
    "ALERT"
  ],
  "max_instances": 0,
  "noteworthy": false,
  "hidden": false,
  "capabilities": [
    "Alert Destination"
  ],
  "connector_count": 0,
  "group_order": 3
},
{
  "type": "SYSLOG",
  "name": "Syslog",
  "desc": "Send alerts to Syslog server",
  "long_desc": "Send Tetration Alerts to Syslog server.",
  "group": "Alerts",
  "index": 0,
  "appliance_type": "tetration_edge",
  "state": "disabled",
  "limit_per_appliance": 1,
  "limit_per_rootscope": 1,
  "limit_per_cluster": 150,
  "config": [
    "SYSLOG_NOTIFIER",
    "SYSLOG_SEVERITY_MAPPING",
    "ALERT"
  ],
  "max_instances": 0,
  "noteworthy": false,
  "hidden": false,
  "capabilities": [
    "Alert Destination"
  ],
  "connector_count": 0,
  "group_order": 3
},
{
  "type": "EMAIL",
  "name": "Email",
  "desc": "Send alerts to Email",
  "long_desc": "Send Tetration Alerts to Email.",
  "group": "Alerts",
  "index": 1,
  "appliance_type": "tetration_edge",
  "state": "disabled",
  "limit_per_appliance": 1,
  "limit_per_rootscope": 1,
  "limit_per_cluster": 150,
  "config": [
    "EMAIL_NOTIFIER",
    "ALERT"
  ],
  "max_instances": 0,
  "noteworthy": false,
  "hidden": false,
  "capabilities": [
```

```
      "Alert Destination"
    ],
    "connector_count": 0,
    "group_order": 3
  },
  {
    "type": "PAGERDUTY",
    "name": "Pager Duty",
    "desc": "Send alerts to Pager Duty",
    "long_desc": "Send Tetration Alerts to Pager Duty",
    "group": "Alerts",
    "index": 3,
    "appliance_type": "tetration_edge",
    "state": "disabled",
    "limit_per_appliance": 1,
    "limit_per_rootscope": 1,
    "limit_per_cluster": 150,
    "config": [
      "PAGERDUTY_NOTIFIER",
      "ALERT"
    ],
    "max_instances": 0,
    "noteworthy": false,
    "hidden": false,
    "capabilities": [
      "Alert Destination"
    ],
    "connector_count": 0,
    "group_order": 3
  },
  {
    "type": "ISE",
    "name": "ISE",
    "desc": "Collect endpoints and inventories from Cisco ISE",
    "long_desc": "Collect information about endpoints and inventories managed by Cisco
ISE appliances and enrich endpoint inventories with user attributes and secure group
tags (SGT). Recommended usage with Cisco Secure Firewall connector for flow visibility.",

    "group": "Endpoints",
    "index": 1,
    "appliance_type": "tetration_edge",
    "state": "disabled",
    "limit_per_appliance": 1,
    "limit_per_rootscope": 1,
    "limit_per_cluster": 150,
    "config": [
      "LDAP",
      "LOG",
      "ALERT"
    ],
    "instance_spec": "ISE",
    "max_instances": 20,
    "noteworthy": false,
    "hidden": false,
    "capabilities": [
      "User Insights",
      "Inventory Enrichment",
      "Endpoint Insights",
      "Software Compliance Posture",
      "MDM Insights"
    ],
    "connector_count": 0,
    "group_order": 1
  },
```

```
 {
   "type": "MERAKI",
   "name": "Meraki",
   "desc": "Collect telemetry from Meraki firewalls",
   "long_desc": "Collect telemetry data from Meraki firewalls.",
   "group": "Flow Ingest",
   "index": 5,
   "appliance_type": "tetration_data_ingest",
   "state": "disabled",
   "limit_per_appliance": 1,
   "limit_per_rootscope": 10,
   "limit_per_cluster": 100,
   "config": [
     "LOG",
     "ALERT"
   ],
   "max_instances": 0,
   "noteworthy": false,
   "hidden": false,
   "capabilities": [
     "Flow Visibility"
   ],
   "connector_count": 0,
   "group_order": 0
 },
 {
   "type": "SERVICENOW",
   "name": "ServiceNow",
   "desc": "Collect ServiceNow CMDB records for inventories",
   "long_desc": "Collect CMDB information and service records from ServiceNow instance
and enriches endpoint inventories with cmdb attributes.",
   "group": "Inventory Enrichment",
   "index": 1,
   "appliance_type": "tetration_edge",
   "state": "disabled",
   "limit_per_appliance": 1,
   "limit_per_rootscope": 1,
   "limit_per_cluster": 150,
   "config": [
     "SERVICENOW",
     "SYNC_INTERVAL",
     "LOG",
     "ALERT"
   ],
   "instance_spec": "SERVICENOW",
   "max_instances": 10,
   "noteworthy": false,
   "hidden": false,
   "capabilities": [
     "User Insights",
     "Inventory Enrichment",
     "Endpoint Insights",
     "Software Compliance Posture"
   ],
   "connector_count": 0,
   "group_order": 2
 },
 {
   "type": "ERSPAN",
   "name": "ERSPAN",
   "desc": "Collect ERSPAN traffic",
   "long_desc": "",
   "group": "Flow Ingest",
   "index": 7,
```

```
  "appliance_type": "tetration_erspan",
  "state": "enabled",
  "limit_per_appliance": 3,
  "limit_per_rootscope": 24,
  "limit_per_cluster": 450,
  "config": [],
  "max_instances": 0,
  "noteworthy": false,
  "hidden": false,
  "capabilities": [],
  "connector_count": 3,
  "group_order": 0
},
{
  "type": "AWS_CONNECTOR",
  "name": "AWS",
  "desc": "AWS Connector",
  "long_desc": "",
  "group": "Cloud",
  "index": 0,
  "appliance_type": "tetration_internal",
  "state": "disabled",
  "limit_per_appliance": 5000,
  "limit_per_rootscope": 5000,
  "limit_per_cluster": 100000,
  "config": [
    "AWS_CONNECTOR"
  ],
  "max_instances": 0,
  "noteworthy": true,
  "pre_release_tag": "BETA",
  "hidden": false,
  "capabilities": [
    "Flow Visibility",
    "Segmentation",
    "Managed K8s",
    "Inventory Enrichment"
  ],
  "connector_count": 0,
  "group_order": 5
},
{
  "type": "AZURE_CONNECTOR",
  "name": "Azure",
  "desc": "Azure Connector",
  "long_desc": "",
  "group": "Cloud",
  "index": 1,
  "appliance_type": "tetration_internal",
  "state": "disabled",
  "limit_per_appliance": 5000,
  "limit_per_rootscope": 5000,
  "limit_per_cluster": 100000,
  "config": [
    "AZURE_CONNECTOR"
  ],
  "max_instances": 0,
  "noteworthy": true,
  "pre_release_tag": "BETA",
  "hidden": false,
  "capabilities": [
    "Flow Visibility",
    "Segmentation",
    "Managed K8s",
```

```
      "Inventory Enrichment"
    ],
    "connector_count": 0,
    "group_order": 5
  },
  {
    "type": "GCP_CONNECTOR",
    "name": "GCP",
    "desc": "GCP Connector",
    "long_desc": "",
    "group": "Cloud",
    "index": 2,
    "appliance_type": "tetration_internal",
    "state": "disabled",
    "limit_per_appliance": 5000,
    "limit_per_rootscope": 5000,
    "limit_per_cluster": 100000,
    "config": [
      "GCP_CONNECTOR"
    ],
    "max_instances": 0,
    "noteworthy": true,
    "pre_release_tag": "BETA",
    "hidden": false,
    "capabilities": [
      "Managed K8s"
    ],
    "connector_count": 0,
    "group_order": 5
  }]
```

コネクタを削除する API

このエンドポイントは、指定されたコネクタを削除します。

```
DELETE /openapi/v1/connectors/<id>
```

ここで、<id> はコネクタを取得する API （1278 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します

応答オブジェクト：削除されたコネクタのステータスを返します。

サンプル応答

```
resp = restclient.delete('/connectors/63c12e316419d0131767e21c')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
 {
    "status": 200,
    "code": 1000,
    "message": "deleted"
  }
```

ID でコネクタを取得する API

このエンドポイントは、指定された ID を持つコネクタを取得します。

```
GET /openapi/v1/connectors/<id>
```

ここで、<id> はコネクタを取得する API （1278 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します

応答オブジェクト：指定された ID を持つコネクタを返します。

サンプル応答

```
resp = restclient.get('/connectors/63c12e316419d0131767e21b')
 if resp.status_code == 200:
   parsed_resp = json.loads(resp.content)
   print json.dumps(parsed_resp)
```

サンプル応答

```
{
   "name": "ERSPAN",
   "updated_at": 0,
   "created_at": 1673604657,
   "appliance_id": "63c1272039042a1c0ddd3e20",
   "root_scope_id": "63bf8d2f497d4f7287dbd335",
   "vrf_id": 1,
   "type": "ERSPAN",
   "ip_bindings": [],
   "internal": false,
   "id": "63c12e316419d0131767e21b"
 }
```

コネクタ名を変更する API

このエンドポイントは、指定された ID を持つコネクタの名前を変更します。

```
PUT /openapi/v1/connectors/<id>
```

ここで、<id> はコネクタを取得する API （1278 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します
name	string	コネクタの新しい名前を指定します

応答オブジェクト：指定された ID を持つコネクタと新しい名前を返します。

サンプル応答

```
req_payload = {
    "name": "ERSPAN2",
```

```
  }
  resp = restclient.put('/ext_appliances/63c12e316419d0131767e21b',
json_body=json.dumps(req_payload))
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "name": "ERSPAN2",
    "updated_at": 0,
    "created_at": 1673604657,
    "appliance_id": "63c1272039042a1c0ddd3e20",
    "root_scope_id": "63bf8d2f497d4f7287dbd335",
    "vrf_id": 1,
    "type": "ERSPAN",
    "ip_bindings": [],
    "internal": false,
    "id": "63c12e316419d0131767e21b"
  }
```

詳細を含むコネクタ情報を取得する API

このエンドポイントは、詳細を含むコネクタ情報を取得します。

```
GET /openapi/v1/connectors/cards/<type>?root_scope_id=<root_scope_id>
```

ここで、<root_scope_id> は範囲の取得 （1122 ページ）の API から取得できる root_scope_id です。リクエストペイロードで、<type> はコネクタタイプを決定する文字列です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲を指定します
type	string	コネクタタイプを指定します[redacted]のいずれかの値を使用できます

応答オブジェクト：指定された範囲のコネクタを返します。

サンプル応答

```
resp = restclient.get('/connectors/cards/NETFLOW?root_scope_id=63bf8d2f497d4f7287dbd335')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "type": "NETFLOW",
    "name": "NetFlow",
    "desc": "Collect telemetry from network devices",
    "long_desc": "Collect NetFlow V9 and/or IPFIX telemetry from network devices such
as routers and switches.",
    "group": "Flow Ingest",
```

```
    "index": 0,
    "appliance_type": "tetration_data_ingest",
    "state": "disabled",
    "limit_per_appliance": 3,
    "limit_per_rootscope": 10,
    "limit_per_cluster": 100,
    "config": [
      "LOG",
      "ALERT"
    ],
    "max_instances": 0,
    "noteworthy": false,
    "hidden": false,
    "capabilities": [
      "Flow Visibility"
    ],
    "connector_count": 0,
    "info": {
      "help": "NetFlow connector collects telemetry data from various network devices
(such as routers, switches).<br> It supports ingest of telemetry data in IPFIX and NetFlow
 V9 protocols. This connector can be used to discover inventory as it provides a network
 context. The connector helps convert data from flow exports and send them securely as
Tetration Flow records into an instance of Tetration. <br><br><b> Connector Alerts:
</b><br> When Connector Alerts are enabled, you may receive the following alerts: <br>
  1. NetFlow Connector is down (due to missing heartbeats). <br>   2. Informational alert
 on high CPU/Memory usage."
    },
    "group_order": 0
  }
```


コネクタを取得する API

このエンドポイントは、指定されたアプライアンスのすべてのコネクタを返します。

```
GET
/openapi/v1/connectors?root_scope_id=<root_scope_id>&appliance_id=<appliance_id>&type=<type>&state=<state>
```

ここで、<root_scope_id> は範囲の取得 （1122 ページ）の API から取得できる root_scope_id です。<appliance_id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる appliance_id です。<type> はアプライアンススキーマを取得する API （1261 ページ）の "connectors" フィールドから取得できるコネクタタイプを決定する文字列です。<state> はコネクタの状態のフィルタです。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲を指定します
appliance_id	string	アプライアンス ID を指定します

名前	タイプ	説明
type	string	コネクタタイプを指定します。“NETFLOW”、“NETSCALER”、“F5”、“AWS”、“ANYCONNECT”、“ASA”、“SLACK”、“KINESIS”、“SYSLOG”、“EMAIL”、“MERAKI”、“PAGERDUTY”、“ISE”、“SERVICENOW”、“ERSPAN”、“AWS_CONNECTOR”、“AZURE_CONNECTOR”、“GCP_CONNECTOR” のいずれかの値を使用できます
state	string	コネクタの状態をフィルタします。“ENABLED”、“DISABLED”、または “UNAVAILABLE”のいずれかの値を使用できます

応答オブジェクト：指定されたアプライアンスのすべてのコネクタを返します。

サンプル応答

```
resp =
restclient.get('root_scope_id=63bf8d2f497d4f7287dbd335&appliance_id=63c1272039042a1c0ddd3e20&type=ERSPAN&state=ENABLED')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
[
    {
      "name": "ERSPAN",
      "updated_at": 0,
      "created_at": 1673604657,
      "appliance_id": "63c1272039042a1c0ddd3e20",
      "root_scope_id": "63bf8d2f497d4f7287dbd335",
      "vrf_id": 1,
      "type": "ERSPAN",
      "ip_bindings": [],
      "state": "enabled",
      "internal": false,
      "id": "63c12e316419d0131767e21b"
    },
    {
      "name": "ERSPAN",
      "updated_at": 0,
      "created_at": 1673604657,
      "appliance_id": "63c1272039042a1c0ddd3e20",
      "root_scope_id": "63bf8d2f497d4f7287dbd335",
```

```
    "vrf_id": 1,
    "type": "ERSPAN",
    "ip_bindings": [],
    "state": "enabled",
    "internal": false,
    "id": "63c12e316419d0131767e21c"
  },
  {
    "name": "ERSPAN",
    "updated_at": 0,
    "created_at": 1673604657,
    "appliance_id": "63c1272039042a1c0ddd3e20",
    "root_scope_id": "63bf8d2f497d4f7287dbd335",
    "vrf_id": 1,
    "type": "ERSPAN",
    "ip_bindings": [],
    "state": "enabled",
    "internal": false,
    "id": "63c12e316419d0131767e21d"
  }
]
```


コネクタを作成する API

このエンドポイントは、指定されたアプライアンスのコネクタを作成します。

```
POST /openapi/v1/connectors
```

リクエストペイロードで、<root_scope_id> は範囲の取得（1122 ページ）の API から取得できる root_scope_id です。<appliance_id> はアプライアンスのリストを取得する API（1251 ページ）から取得できる appliance_id です。<type> はアプライアンススキーマを取得する API（1261 ページ）の "connectors" フィールドから取得できるコネクタタイプを決定する文字列です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
name	string	名前を指定します
root_scope_id	string	ルート範囲を指定します
appliance_id	string	アプライアンス ID を指定します

名前	タイプ	説明
type	string	コネクタタイプを指定します。“NETFLOW”、“NETSCALER”、“F5”、“AWS”、“ANYCONNECT”、“ASA”、“SLACK”、“KINESIS”、“SYSLOG”、“EMAIL”、“MERAKI”、“PAGERDUTY”、“ISE”、“SERVICENOW”、“ERSPAN”、“AWS_CONNECTOR”、“AZURE_CONNECTOR”、“GCP_CONNECTOR” のいずれかの値を使用できます
version	string	バージョンを指定します

応答オブジェクト：作成されたコネクタを返します。

サンプル応答

```
req_payload = {
    "root_scope_id": "63c41ff2497d4f5f5be73662",
    "appliance_id": "63c6ef42bca44e2b5e729191",
    "type": "NETFLOW",
    "name": "netflowtest",
    "version": "1.1.1"
  }
  resp = restclient.post('/connectors', json_body=json.dumps(req_payload))
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "name": "netflowtest",
    "updated_at": 0,
    "created_at": 1674187875,
    "appliance_id": "63c6ef42bca44e2b5e729191",
    "root_scope_id": "63c41ff2497d4f5f5be73662",
    "vrf_id": 1,
    "type": "NETFLOW",
    "ip_bindings": [],
    "sources": [],
    "internal": false,
    "id": "63ca14631a49bd4c0e0cefa2"
  }
```

コネクタ設定タイプでの設定を取得する API

このエンドポイントは、指定された ID を持つコネクタの設定を取得します。

```
GET /openapi/v1/connectors/<id>/config?config_type=<config_type>
```

ここで、<id> はアプライアンスのリストを取得する API （1251 ページ）から取得できる id です。<config_type> はアプライアンススキーマを取得する API （1261 ページ）から取得できる "valid_config" です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します
config_type	string	設定タイプを指定します。使用できるすべての値については、設定のスキーマを取得する API （1248 ページ）の説明にリストされている値を参照してください

応答オブジェクト：指定されたコネクタ ID と設定タイプを持つ設定を返します。

サンプル応答

```
resp = restclient.get('/connectors/63db5418e6ee1167a4c0986c/config?config_type=LOG')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答


```
 [{
    "mode": "TEST_AND_APPLY",
    "name": "Log instance 2/1/23 22:29",
    "root_scope_id": "63d98f45497d4f53005b24fa",
    "vrf_id": 1,
    "appliance_id": "63dad690e6ee1131f255e985",
    "connector_id": "63db5418e6ee1167a4c0986c",
    "service_id": "63db5418e6ee1167a4c0986d",
    "deleted": false,
    "type": "LOG",
    "state": "COMMIT",
    "error_code": "NO_ERROR",
    "error_text": "",
    "attempts": 1,
    "config": {
      "secured": {},
      "unsecured": {
        "log-level": "info",
        "max-log-size": 10,
        "max-log-age": 30,
        "max-log-backups": 20
      }
    },
    "push_to_dio_at": 1675319360,
    "created_at": 1675319360,
    "updated_at": 1675319364,
    "discovery_completed_at": 0,
    "committed_at": 1675319364,
    "delete_at": 0,
    "error_at": 0,
    "hidden": false,
    "discovery_phase": 0,
```

```
    "internal": false,
    "id": "63db5840f029813659f9fcf5"
  }]
```

新しい設定をコネクタに追加する API

このエンドポイントは、指定された ID を持つコネクタに新しい設定を追加します。

```
POST /openapi/v1/connectors/<id>/config
```

ここで、<id> はコネクタを取得する API（1278 ページ）から取得できる id です。<config_type> はアプライアンススキーマを取得する API（1261 ページ）から取得できる "valid_config" です。<config> スキーマを取得するには、すべてのタイプのコネクタを取得する API（1267 ページ）の応答からいずれかの "config" を選択し、設定のスキーマを取得する API（1248 ページ）に "config" を適用します。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
name	string	設定名を指定します
type	string	設定タイプを指定します。使用できるすべての値については、設定のスキーマを取得する API（1248 ページ）の説明にリストされている値を参照してください
config	set	入力された設定スキーマを JSON 形式で提供します

応答オブジェクト：更新された設定を返します。

サンプル応答

```
req_payload = {
    "name": "Log instance 2/1/23 22:29",
    "type": "LOG",
    "config": {
      "secured": {},
      "unsecured": {
        "log-level": "info",
        "max-log-size": 10,
        "max-log-age": 30,
        "max-log-backups": 20
      }
    }
  }
  resp = restclient.post('/connectors/63db5418e6ee1167a4c0986c/config',
json_body=json.dumps(req_payload))
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "mode": "TEST_AND_APPLY",
    "name": "Log instance 2/1/23 11:29",
    "root_scope_id": "63d98f45497d4f53005b24fa",
    "vrf_id": 1,
    "appliance_id": "63dad690e6ee1131f255e985",
    "connector_id": "63db5418e6ee1167a4c0986c",
    "deleted": false,
    "type": "LOG",
    "state": "PENDING",
    "attempts": 0,
    "config": {
      "secured": {},
      "unsecured": {
        "log-level": "info",
        "max-log-size": 10,
        "max-log-age": 30,
        "max-log-backups": 20
      }
    },
    "push_to_dio_at": 0,
    "created_at": 1675322272,
    "updated_at": 1675322272,
    "discovery_completed_at": 0,
    "committed_at": 0,
    "delete_at": 0,
    "error_at": 0,
    "hidden": false,
    "discovery_phase": 0,
    "internal": false,
    "id": "63db63a0f029813659f9fcf7"
  }
```


設定を削除する API

このエンドポイントは、指定されたコネクタから設定を削除します。

```
DELETE /openapi/v1/connectors/<id>/config/<config_id>
```

ここで、<id>はコネクタを取得する API（1278 ページ）から取得できる id です。<config_id>はコネクタ設定タイプでの設定を取得する API（1281 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します
config_id	string	設定 ID を指定します

応答オブジェクト：指定されたコネクタの削除された設定のステータスを返します。

サンプル応答

```
resp =
restclient.delete('/connectors/63c1272039042a1c0ddd3e20/config/63c1272039042a1c0ddd3e21')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "status": 200,
    "code": 1000,
    "message": "deleted"
  }
```

-

設定を取得する API

このエンドポイントは、指定された ID を持つ設定を取得します。

```
GET /openapi/v1/connectors/<id>/config/<config_id>
```

ここで、<id> はコネクタを取得する API（1278 ページ）から取得できる id です。<config_id> はコネクタ設定タイプでの設定を取得する API（1281 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	アプライアンス ID を指定します
config_id	string	設定 ID を指定します

応答オブジェクト：指定された ID を持つ設定を返します。

サンプル応答

```
resp =
restclient.get('/connectors/63db5418e6ee1167a4c0986c/config/63db5840f029813659f9fcf5')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "mode": "TEST_AND_APPLY",
    "name": "Log instance 2/1/23 22:29",
    "root_scope_id": "63d98f45497d4f53005b24fa",
    "vrf_id": 1,
    "appliance_id": "63dad690e6ee1131f255e985",
    "connector_id": "63db5418e6ee1167a4c0986c",
    "service_id": "63db5418e6ee1167a4c0986d",
    "deleted": false,
    "type": "LOG",
    "state": "COMMIT",
    "error_code": "NO_ERROR",
    "error_text": "",
    "attempts": 1,
    "config": {
      "secured": {},
      "unsecured": {
        "log-level": "info",
        "max-log-size": 10,
        "max-log-age": 30,
        "max-log-backups": 20
      }
    },
    "push_to_dio_at": 1675319360,
```

```
    "created_at": 1675319360,
    "updated_at": 1675319364,
    "discovery_completed_at": 0,
    "committed_at": 1675319364,
    "delete_at": 0,
    "error_at": 0,
    "hidden": false,
    "discovery_phase": 0,
    "internal": false,
    "id": "63db5840f029813659f9fcf5"
  }
```


このコネクタで使用可能なすべてのトラブルシューティング コマンドを表示する API

このエンドポイントは、指定されたコネクタで使用できるトラブルシューティング コマンドのリストを返します。

```
GET /openapi/v1/connectors/<id>/commands/available
```

ここで、<id> はコネクタを取得する API （1278 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します

応答オブジェクト：指定されたコネクタで使用できるトラブルシューティング コマンドのリストを返します。

サンプル応答

```
resp = restclient.get('/connectors/63c6f2701a49bd2bb0696cab/commands/available')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答


```
[
    {
      "type": "LIST_SERVICE_FILES",
      "name": "List a directory in a service"
    },
    {
      "type": "CONTROLLER_PROFILING",
      "name": "Collect controller profile"
    },
    {
      "type": "SHOW_LOG",
      "name": "Show logs"
    },
    {
      "type": "SHOW_SERVICE_LOG",
      "name": "Show service logs"
    },
    {
      "type": "RESTART_CONNECTOR_CONTAINER",
      "name": "Restart the connector docker container"
    },
    {
      "type": "SHOW_SUPERVISOR_COMMANDS",
```

```
    "name": "Execute supervisorctl command"
  },
  {
    "type": "CONNECTOR_ALERT_INTERVAL_CONNECTOR",
    "name": "Override connector alert interval for Connector"
  },
  {
    "type": "SERVICE_PROFILING",
    "name": "Collect connector profile"
  },
  {
    "type": "SNAPSHOT_CONNECTOR",
    "name": "Collect Snapshot from a connector"
  },
  {
    "type": "PACKET_CAPTURE",
    "name": "Packet capture"
  },
  {
    "type": "NETWORK_CONNECTIVITY_COMMANDS",
    "name": "Test network connectivity"
  },
  {
    "type": "SHOW_SERVICE_RUNNING_CONF",
    "name": "Show running configuration of a service"
  },
  {
    "type": "RESTART_CONNECTOR_SERVICE",
    "name": "Restart the connector service"
  },
  {
    "type": "SHOW_SYS_COMMANDS",
    "name": "Execute system command"
  }
]
```

すべてのトラブルシューティング コマンドを表示する API

このエンドポイントは、指定されたコネクタに対して有効になっているトラブルシューティング コマンドのリストを返します。

```
GET /openapi/v1/connectors/<id>/commands
```

ここで、<id> はコネクタを取得する API （1278 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します

応答オブジェクト：指定されたコネクタに対して有効になっているトラブルシューティング コマンドのリストを返します。

サンプル応答

```
resp = restclient.get('/connectors/63db5418e6ee1167a4c0986c/commands')
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
[
    {
      "appliance_id": "63dad690e6ee1131f255e985",
      "connector_id": "63db5418e6ee1167a4c0986c",
      "service_id": "63db5418e6ee1167a4c0986d",
      "state": "success",
      "level": "SERVICE",
      "command": "SHOW_LOG",
      "arg_string": "",
      "args": {
        "pattern": "info"
      },
      "tailed": false,
      "rc": 0,
      "push_to_dio_at": 1675319615,
      "attempts": 1,
      "error_code": "NO_ERROR",
      "error_text": "",
      "deleted": false,
      "deleted_at": 0,
      "created_at": 1675319613,
      "total_chunk": 0,
      "id": "63db593df029813659f9fcf6"
    }
  ]
```


トラブルシューティング コマンドを作成する API

このエンドポイントは、指定されたコネクタで使用できるトラブルシューティング コマンドを作成します。

```
POST /openapi/v1/connectors/<id>/commands
```

ここで、<id> はコネクタを取得する API（1278 ページ）から取得できる id です。リクエストペイロードで、<command> はこのコネクタで使用可能なすべてのトラブルシューティング コマンドを表示する API（1286 ページ）から取得できるトラブルシューティング コマンド タイプです。<arguments> はコマンドスキーマの入力された JSON オブジェクトであり、トラブルシューティング コマンドのスキーマを取得する API（1250 ページ）から取得できます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します
command	string	コマンドタイプを指定します
引数	set	入力されたコマンドスキーマを JSON 形式で提供します

応答オブジェクト：指定されたアプライアンスの作成されたトラブルシューティング コマンドを返します。

サンプル応答

```
req_payload = {
    "command": "SHOW_LOG",
    "arguments": {
      "pattern": "info"
```

```
    }
  }
  resp = restclient.post('/connectors/63db5418e6ee1167a4c0986c/commands',
json_body=json.dumps(req_payload))
  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "appliance_id": "63dad690e6ee1131f255e985",
    "connector_id": "63db5418e6ee1167a4c0986c",
    "service_id": "63db5418e6ee1167a4c0986d",
    "state": "pending",
    "level": "SERVICE",
    "command": "SHOW_LOG",
    "args": {
      "pattern": "info"
    },
    "tailed": false,
    "rc": 0,
    "push_to_dio_at": 0,
    "attempts": 0,
    "deleted": false,
    "deleted_at": 0,
    "created_at": 1675319613,
    "total_chunk": 0,
    "id": "63db593df029813659f9fcf6"
  }
```

トラブルシューティング コマンドを削除する API

このエンドポイントは、指定されたコネクタで使用できるトラブルシューティング コマンドを削除します。

```
DELETE /openapi/v1/connectors/<id>/commands/<command_id>
```

ここで、<id>はコネクタを取得する API（1278 ページ）から取得できる id です。<command_id>はすべてのトラブルシューティング コマンドを表示する API（1263 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します
command_id	string	コマンド ID を指定します

応答オブジェクト：指定されたコネクタの削除されたトラブルシューティング コマンドのステータスを返します。

サンプル応答

```
resp =
restclient.delete('/connectors/63c12e316419d0131767e21c/commands/63c10a0039042a6aee1b008c')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "status": 200,
    "code": 1000,
    "message": "deleted"
  }
```

トラブルシューティング コマンドをルックアップする API

このエンドポイントは、指定されたコネクタに対して選択されたトラブルシューティング コマンドを返します。

```
GET /openapi/v1/connectors/<id>/commands/<command_id>
```

ここで、<id>はコネクタを取得する API（1278 ページ）から取得できる id です。<command_id>はすべてのトラブルシューティング コマンドを表示する API（1263 ページ）から取得できる id です。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します
command_id	string	コマンド ID を指定します

応答オブジェクト：指定されたコネクタに対して選択されたトラブルシューティング コマンドを返します。

サンプル応答

```
resp =
restclient.get('/connectors/63db5418e6ee1167a4c0986c/commands/63db593df029813659f9fcf6')

  if resp.status_code == 200:
    parsed_resp = json.loads(resp.content)
    print json.dumps(parsed_resp)
```

サンプル応答

```
{
    "appliance_id": "63dad690e6ee1131f255e985",
    "connector_id": "63db5418e6ee1167a4c0986c",
    "service_id": "63db5418e6ee1167a4c0986d",
    "state": "success",
    "level": "SERVICE",
    "command": "SHOW_LOG",
    "arg_string": "",
    "args": {
      "pattern": "info"
    },
    "tailed": false,
    "rc": 0,
    "push_to_dio_at": 1675319615,
    "attempts": 1,
    "error_code": "NO_ERROR",
    "error_text": "",
    "deleted": false,
    "deleted_at": 0,
    "created_at": 1675319613,
```

```
    "total_chunk": 0,
    "id": "63db593df029813659f9fcf6"
  }
```

コネクタコマンドの出力をファイルとしてダウンロードする API

このエンドポイントは、コマンドの出力をファイルとしてダウンロードします。

```
GET /openapi/v1/connectors/<id>/commands/{command_id}/download
```

ここで、<id> はコネクタを取得する API（1278 ページ）から取得できる id です。<command_id> はすべてのトラブルシューティング コマンドを表示する API（1263 ページ）から取得できる id です。すべてのコマンドにダウンロード可能な出力があるわけではありません。トラブルシューティング コマンドのスキーマを取得する API（1250 ページ）によって提供されるコマンドスキーマを確認してください。ここで、"output_type": "FILE" はダウンロード可能なコンテンツがあることを示し、"output_ext" はファイルタイプを示しています。

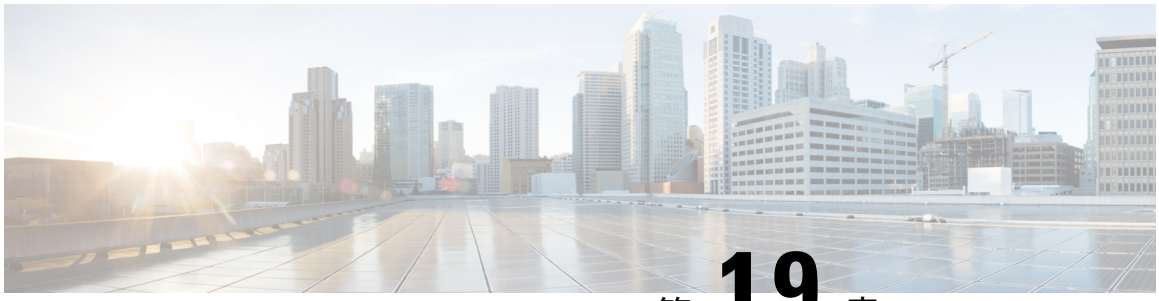
パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
id	string	コネクタ ID を指定します
command_id	string	コマンド ID を指定します

応答オブジェクト：指定されたアプライアンスに対して選択されたトラブルシューティング コマンドを返します。

サンプル応答

```
resp = restclient.download('downloadFile',
'/connectors/63c6ef42bca44e2b5e729191/commands/63cace941a49bd4c0e0cf45a/download')
```



第 19 章

制限

- フローとエンドポイント （1293 ページ）
- テナント、子範囲、インベントリフィルタ、およびロール （1293 ページ）
- Cloud Connector （1294 ページ）
- コネクタ （1295 ページ）
- ラベルの制限 （1297 ページ）
- ポリシーに関連する制限 （1298 ページ）
- その他の機能 （1300 ページ）
- データイン/データアウト （1305 ページ）

フローとエンドポイント

メトリック	制限	8RU/39RU/SaaS/-
Cisco Secure Workload によるテレメトリデータの同時分析が可能なサーバー数（仮想マシン、ベアメタル）	最大 5000 *	8RU
	最大 25000 *	39RU
Secure Workload の 1 秒あたりの処理可能フローイベント数	1 秒あたり最大 50000	8RU
	1 秒あたり最大 200 万	39RU

テナント、子範囲、インベントリフィルタ、およびロール

メトリック	制限	8RU/39RU
テナント数	7	8RU
	35	39RU

メトリック	制限	8RU/39RU
テナントあたりの子範囲の数	1000 *	8RU
	5000	39RU
テナント全体の子範囲の総数	7000	8RU
	35000	39RU
テナントあたりのワークスペースの数	1000 *	8RU
	3500 *	39RU
テナント間のワークスペースの総数	5000	8RU
	20000	39RU
テナントあたりのインベントリフィルタの数	1000 *	8RU
	5000 *	39RU
テナント全体のインベントリフィルタの総数	7000 *	8RU
	35000 *	39RU
子範囲あたりのロール数	6	8RU
	6	39RU

Cloud Connector

Cloud Connector	メトリック	制限	規模	仮想ネットワーク	Kubernetes クラスタ
AWS コネクタ	AWS コネクタによってエクスポートされるフローの総数	毎秒 15000 フロー	コネクタあたり 5 アカウント	アカウントあたり 5	アカウントあたり 5
Azure コネクタ	Azure コネクタによってエクスポートされるフローの総数	毎秒 15000 フロー	コネクタあたり 5 サブスクリプション	サブスクリプションあたり 5	サブスクリプションあたり 5

Cloud Connector	メトリック	制限	規模	仮想ネットワーク	Kubernetes クラスタ
Google Cloud Platform	GCP コネクタによってエクスポートされるフローの総数	毎秒 15000 フロー	コネクタあたり 5 プロジェクト	プロジェクトあたり 5	プロジェクトあたり 5

（注）
- クラスタ内ですべてのテナントに渡って、最大 50 個のコネクタ（クラウドコネクタを含む）を設定できます。
- Cisco Secure Workload のクラウドコネクタによって管理されるワークロードにはワークロードライセンスが必要であるため、ワークロードの合計数がライセンスされていて、クラスタの制限内であることを確認します。

コネクタ

（注）
- クラスタ内ですべてのテナントに渡って、最大 50 個のコネクタ（クラウドコネクタを含む）を設定できます。
- 個々のコネクタに適用される制限については、「コネクタとは」を参照してください。

コネクタ	メトリック	制限
AnyConnect コネクタ	1 つの AnyConnect コネクタでサポートされる AnyConnect エンドポイントの総数	5000 エンドポイント （注） すべての AnyConnect プロキシセンサーにわたる AnyConnect エンドポイントの数は、Secure Workload アプライアンスでサポートされるセンサーの数によって制限されます。

コネクタ	メトリック	制限
AnyConnect コネクタ	AnyConnect エンドポイントのインベントリでラベル付けできる LDAP 属性の数	6 属性
AWS コネクタ	AWS コネクタによってエクスポートされるフローの総数	毎秒 15000 フロー
F5 コネクタ	F5 コネクタによってエクスポートされるフローの総数	毎秒 15000 フロー
NetFlow コネクタ	1 つの NetFlow コネクタによってエクスポートされるフローの総数	毎秒 15000 フロー
NetScaler コネクタ	NetScaler コネクタによってエクスポートされるフローの総数	毎秒 15000 フロー

コネクタ用の Cisco Secure Workload 仮想アプライアンス

アプライアンス	メトリック	制限
Cisco Secure Workload Ingest アプライアンス	1つのアプライアンスにおけるコネクタ数	3
	ルート範囲ごとのアプライアンス数	100
	クラスタごとのアプライアンス数	500
Cisco Secure Workload Edge アプライアンス	1つのアプライアンスにおけるコネクタ数	6
	ルート範囲ごとのアプライアンス数	1
	クラスタごとのアプライアンス数	ルート範囲の数

ラベルの制限

機能	メトリック	制限	8RU/39RU/SaaS/-
ラベルの制限	すべてのルート範囲でラベル付けできる IP アドレスの最大数	1,500,000 *	39RU
		500,000 *	8RU
	すべてのルート範囲でラベル付けできるサブネットの最大数	200,000	39RU
		50,000	8RU
	テナントごとにラベル付けできる IP アドレスの最大数（CMDB のみ）	6,000 / 100 ライセンス（SaaS のみ）	
	テナントごとにラベル付けできるサブネットの最大数（CMDB のみ）	120 / 100 ライセンス（SaaS のみ）	

ポリシーに関連する制限

機能	メトリック	制限	8RU/39RU/SaaS/-
自動ポリシー検出（旧ADM）	単一範囲での自動ポリシー検出の実行で許可されるメンバーワークロード（エンドポイント）の最大数	5000	-
	単一範囲での自動ポリシー検出の実行で許可されるカンバセーションの最大数	10,000,000	-
	範囲ツリーのブランチでの自動ポリシー検出の実行で許可されるメンバーワークロード（エンドポイント）の最大数	25000	-
	範囲ツリーのブランチでの自動ポリシー検出の実行で許可されるカンバセーションの最大数	20,000,000	-
	自動ポリシー検出の実行で許可される合計固有ワークロード（エンドポイント）の最大数	15,000,000	-
	デフォルトポリシー検出設定の除外フィルタの最大数	100	-
	ワークスペースごとに許可される除外フィルタの最大数	100	-

機能	メトリック	制限	8RU/39RU/SaaS/-
具体的なポリシー	Kubernetes 以外のワークロードにインストールされたエージェントでのポリシーの合計サイズ	2.5 MB （複雑さに応じて、約 2000 個のポリシー）	-
	Kubernetes ノードにインストールされたエージェントでのポリシーの合計サイズ	7.5 MB （複雑さに応じて、約 6000 個のポリシー）	-

その他の機能

機能	メトリック	制限	8RU/39RU/SaaS/-
Alerts	ルート範囲内のサポートされるインスタンス数	256	-
	ルート範囲内のサポートされるインスタンス数	1024	-
	ルート範囲ごとの UI に表示される最新のアラート数	5000	-
	UI でプレビューする最大アラートレート		-

機能	メトリック	制限	8RU/39RU/SaaS/-
		1 分あたり 60 件 （注） 1 分あたり 60 件を超えるアラートが送信される場合、UI には、アラートがデータタップに送信されたが UI で抑制されていることを示す概要メッセージが表示されます。1 分あたり 60 件のアラートは、アラートがデータタップに送信されるレートに適用され、アラート時間やイベント時間には適用されず、特定のデータバッチとは関係がないことに注意してくださ	

機能	メトリック	制限	8RU/39RU/SaaS/-
		い。	
	ルート範囲ごとに設定されたアラートの数（モーダル経由）	[1000]	-
	1 分あたりのアラートアプリによってバッチ処理されるアラートの最大数	20000	-
コンプライアンスアプリ	サポートされるワークスペースの数	128	-
Neighborhood アプリ	Neighborhood アプリを有効にできるルート範囲の数	256	-
	ルート範囲ごとのタイプごとのアラート設定の最大数 （注） ルート範囲ごとの Neighborhood アプリの各タイプのアラート設定数が 30 以内であることを確認してください。	30	-
	範囲ごとのライブ分析フィルタとクラスタの最大数	500	-

機能	メトリック	制限	8RU/39RU/SaaS/-
Alerts	ルート範囲内のサポートされるインスタンス数	256	-
	ルート範囲内のサポートされるインスタンス数	1024	-
	ルート範囲ごとの UI に表示される最新のアラート数	5000	-
	UI でプレビューする最大アラートレート		-

機能	メトリック	制限	8RU/39RU/SaaS/-
		1 分あたり 60 件 （注） 1 分あたり 60 件を超えるアラートが送信される場合、UI には、アラートがデータタップに送信されたが UI で抑制されていることを示す概要メッセージが表示されます。1 分あたり 60 件のアラートは、アラートがデータタップに送信されるレートに適用され、アラート時間やイベント時間には適用されず、特定のデータバッチとは関係がないことに注意してくださ	

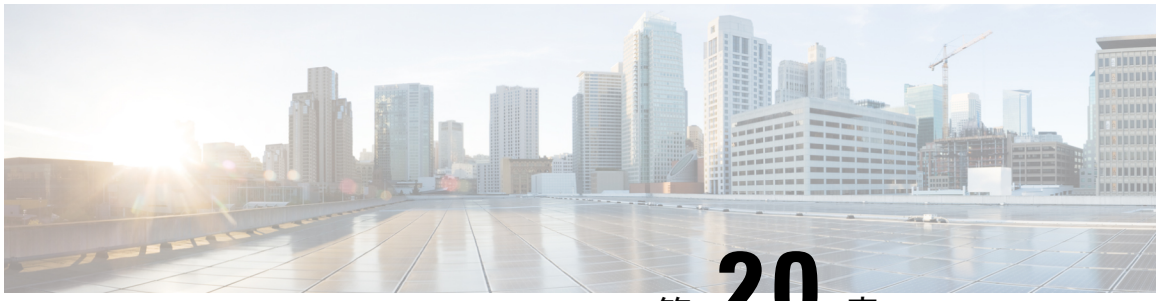
機能	メトリック	制限	8RU/39RU/SaaS/-
		い。	
	ルート範囲ごとに設定されたアラートの数（モーダル経由）	[1000]	-
	1 分あたりのアラートアプリによってバッチ処理されるアラートの最大数	20000	-
コンプライアンスアプリ	サポートされるワークスペースの数	128	-

機能	メトリック	制限	8RU/39RU/-
追跡するインベントリ項目の数	すべてのルート範囲で追跡できる IP アドレスの最大数	1,500,000 *	39RU
		500,000 *	8RU
	すべてのルート範囲で追跡できるサブネットの最大数	200,000	39RU
		50,000	8RU

データタイン/データアウト

機能	メトリック	制限	8RU/39RU/SaaS/-
データタップ	アプライアンスごとにサポートされるデータタップの数	10	-

（注） * すべてのエージェントでカンバセーションモードが有効になっている場合、アスタリスク（*）マークの付いた制限について、Secure Workload は記載されている制限の最大 2 倍をサポートします。詳細については、「カンバセーションモード」を参照してください。



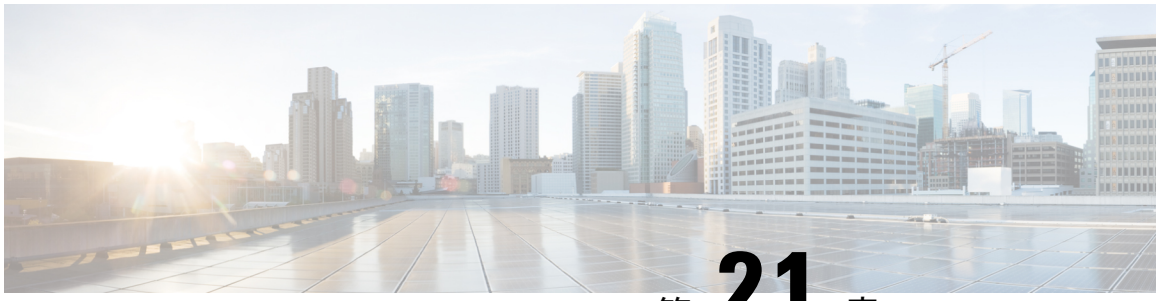
第 20 章

Cisco Secure Workload Virtual （SECURE-WORKLOAD-V）

• Cisco Secure Workload Virtual （1307 ページ）

Cisco Secure Workload Virtual

Secure Workload Virtual（以前の Tetration-V）を展開する手順は、https://www.cisco.com/c/en/us/support/security/tetration-analytics-g1/model.html に掲載されています。



第 21 章

エンド ユーザ ライセンス契約書

• エンド ユーザ ライセンス契約書 （1309 ページ）

エンド ユーザ ライセンス契約書

お使いの製品のエンドユーザーライセンス契約および補足エンドユーザーライセンス契約を表示するには、https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.htmlにアクセスしてください。[補足エンドユーザーライセンス契約（Supplemental End User License Agreements）] タブをクリックして、製品を検索します。

エンド ユーザ ライセンス契約書

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。