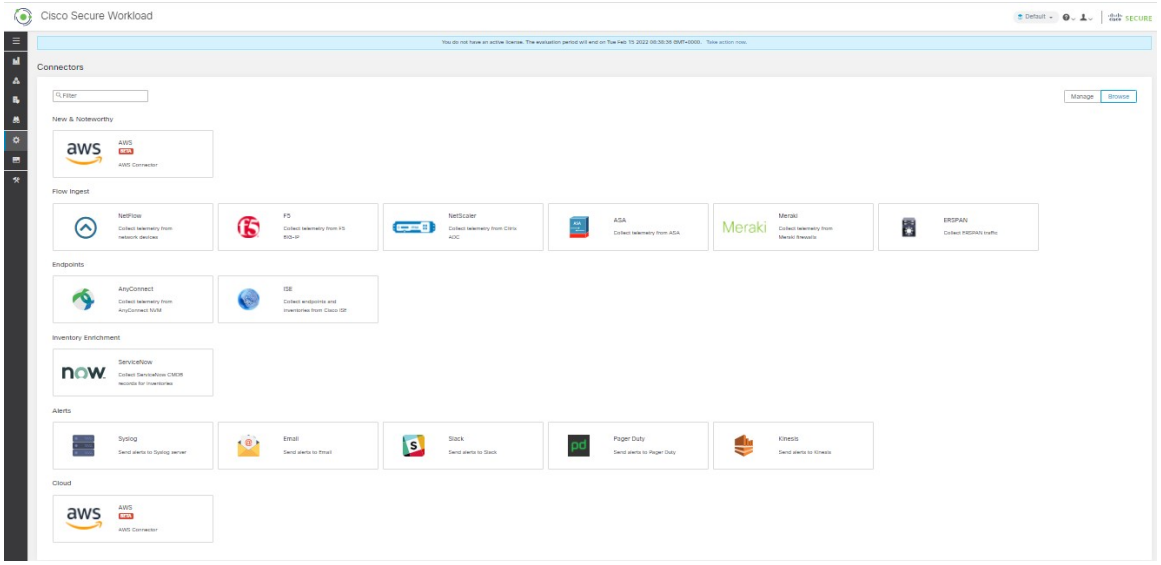
コネクタ

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
- フローの取り込み、インベントリの強化、ポリシー適用のための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 *1:* 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

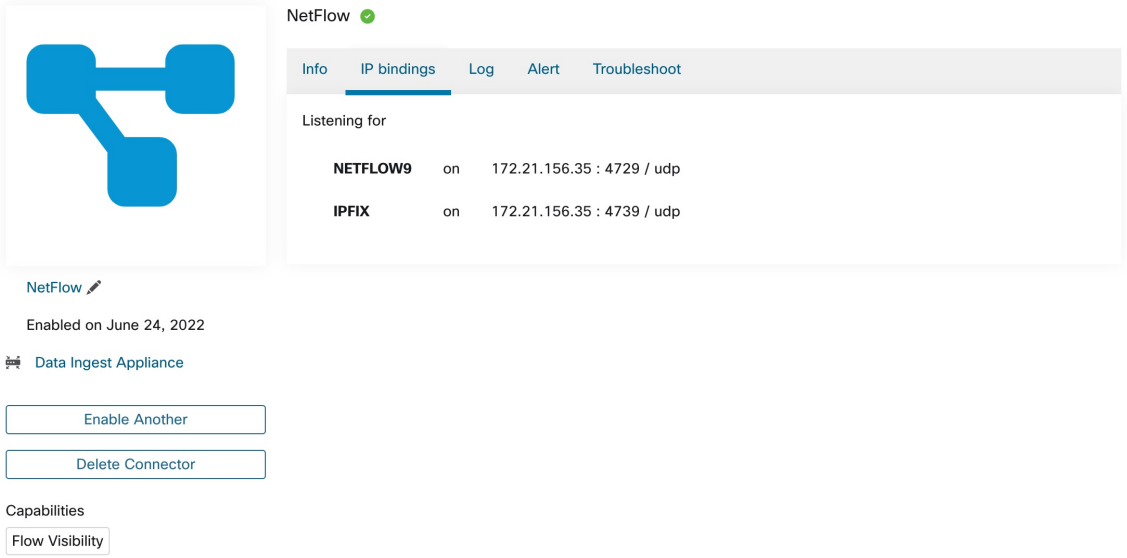
コネクタ	説明	仮想アプライアンス上に展開
F5 BIG-IP	F5 BIG-IP からテレメトリを収集し、クライアント側とサーバー側のフローをつなぎ合わせ、ユーザー属性でクライアントインベントリを充実させます。	Cisco Secure Workload Ingest
Citrix NetScaler	Citrix ADC からテレメトリを収集し、クライアント側とサーバー側のフローをつなぎ合わせます。	Cisco Secure Workload Ingest
Cisco Secure Connector Firewall	Cisco Secure Firewall ASA および Cisco 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 *2 : 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。フォームを使用して、ユーザーに次の情報の提供を求めます。VRF の名前、コネクタの IP サブネット、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。

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。	Yes
2	packetDeltaCount	このフローの着信パケット数。	Yes

Element ID	名前	説明	必須
4	protocolIdentifier	IPパケットヘッダーのプロトコル番号の値。	Yes
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。	対応
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。	対応
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）。	×
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）。	×

Element ID	名前	説明	必須
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。

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 *3 : 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、次の設定が許可されています。

- 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情報を提供することにより、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。

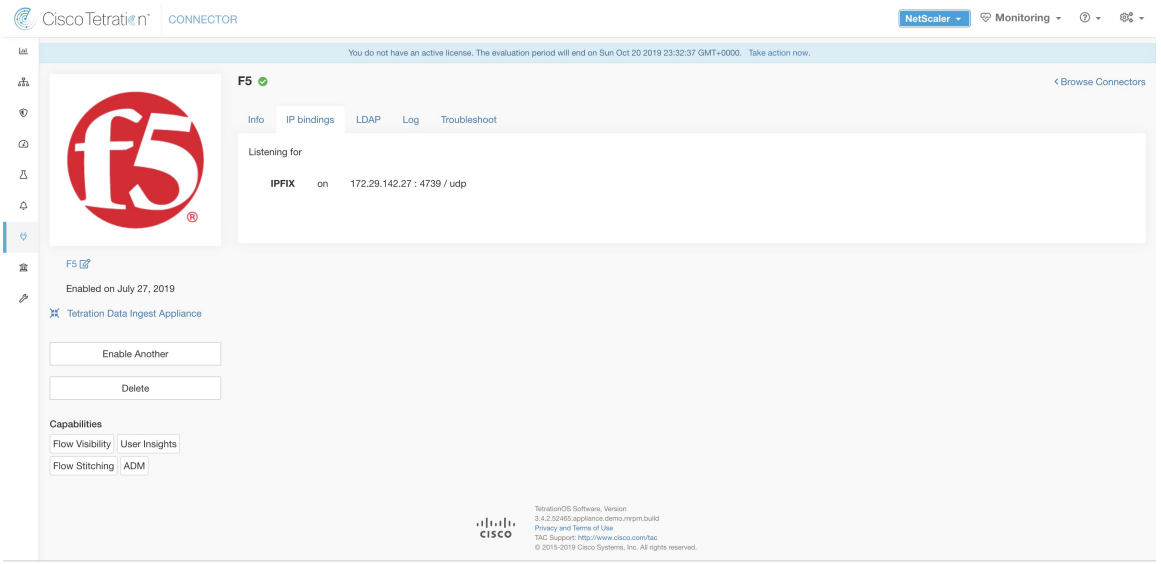
制限

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 *4 : 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 *5 : 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、次の設定が許可されています。

- 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

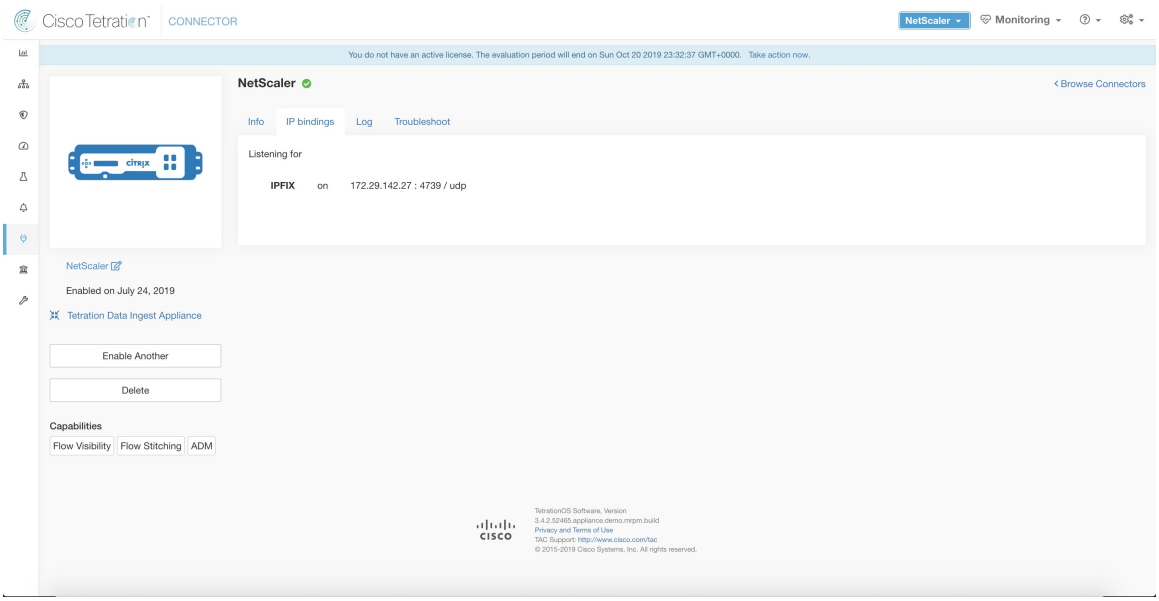
メトリック	制限
1 つの Secure Workload Ingest アプライアンスにおける F5 コネクタの最大数	3

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 *6 : 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）。

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。

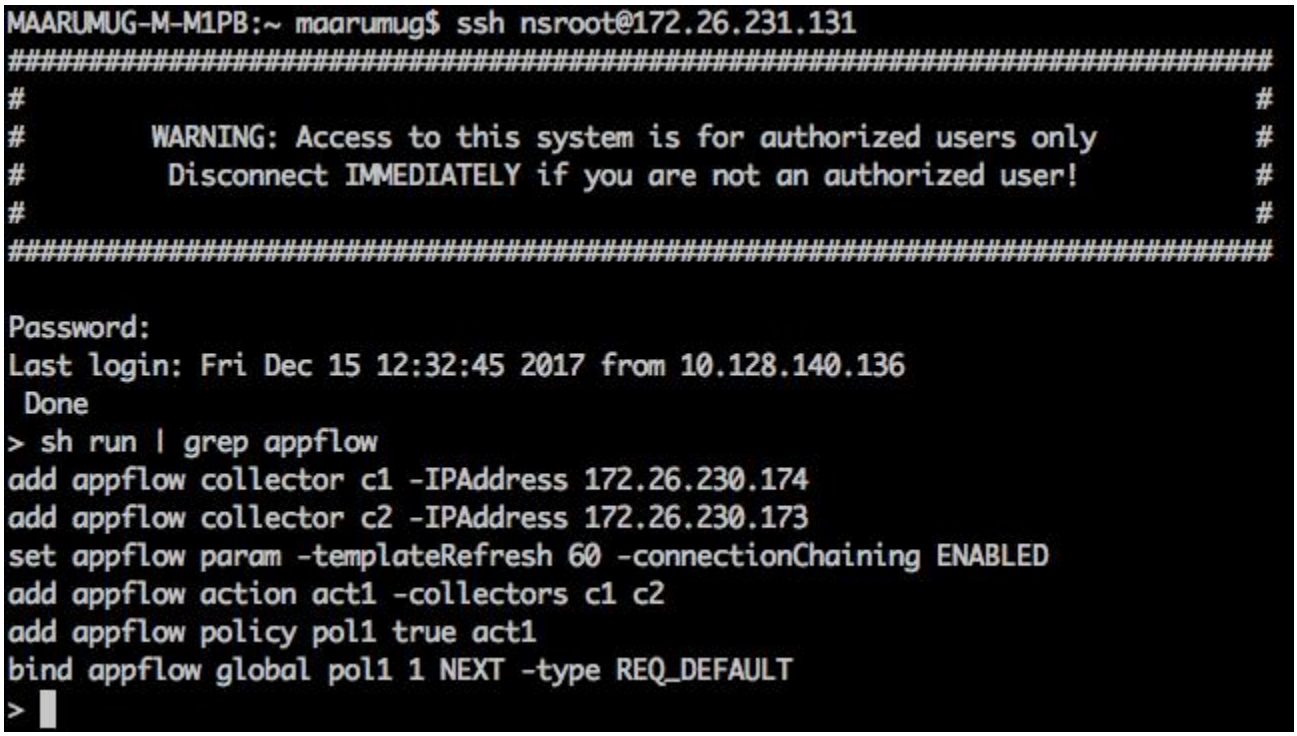
テンプレート更新のデフォルト値は 60 秒です。

```
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 *7: 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、次の構成が許可されています。

- 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情報を提供することにより、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 *1:* 制限

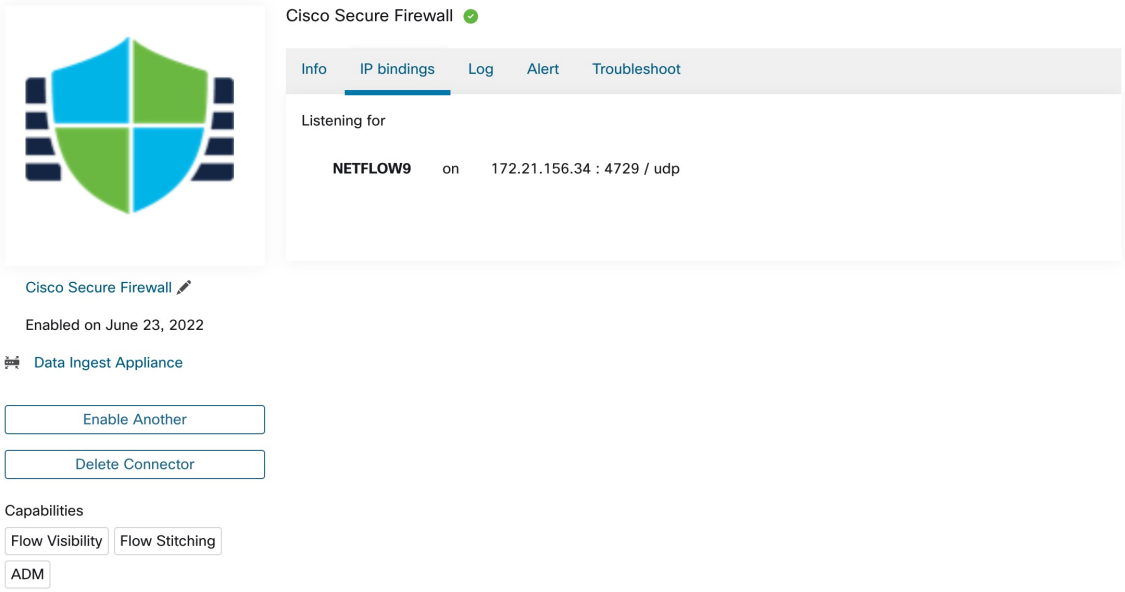
メトリック	制限
1 つの Secure Workload Ingest アプライアンス上の NetScaler コネクタの最大数	3

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 *8 : 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。フォームを使用して、ユーザーに次の情報の提供を求めます。VRF の名前、コネクタの IP サブネット、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。

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、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情報を提供することにより、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。

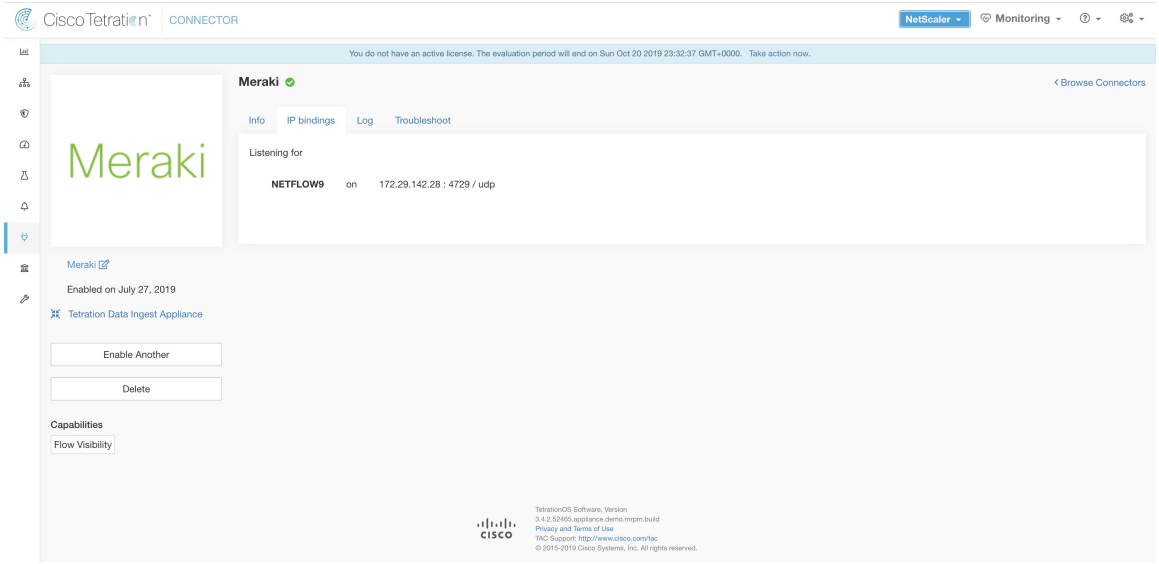
制限

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 *9 : 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。

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。

順方向フロー観測データ

フィールド	Element ID	エレメント名
プロトコル	4	*protocolIdentifier*
Source Address	8	*sourceIPv4Address*
送信元ポート（Source Port）	7	*sourceTransportPort*
宛先アドレス	12	*destinationIPv4Address*
接続先ポート	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

逆方向フロー観察情報

フィールド	Element ID	
プロトコル	4	*protocolIdentifier*
Source Address	8	*sourceIPv4Address*
送信元ポート（Source Port）	7	*sourceTransportPort*

フィールド	Element ID	
宛先アドレス	12	*destinationIPv4Address*
接続先ポート	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。

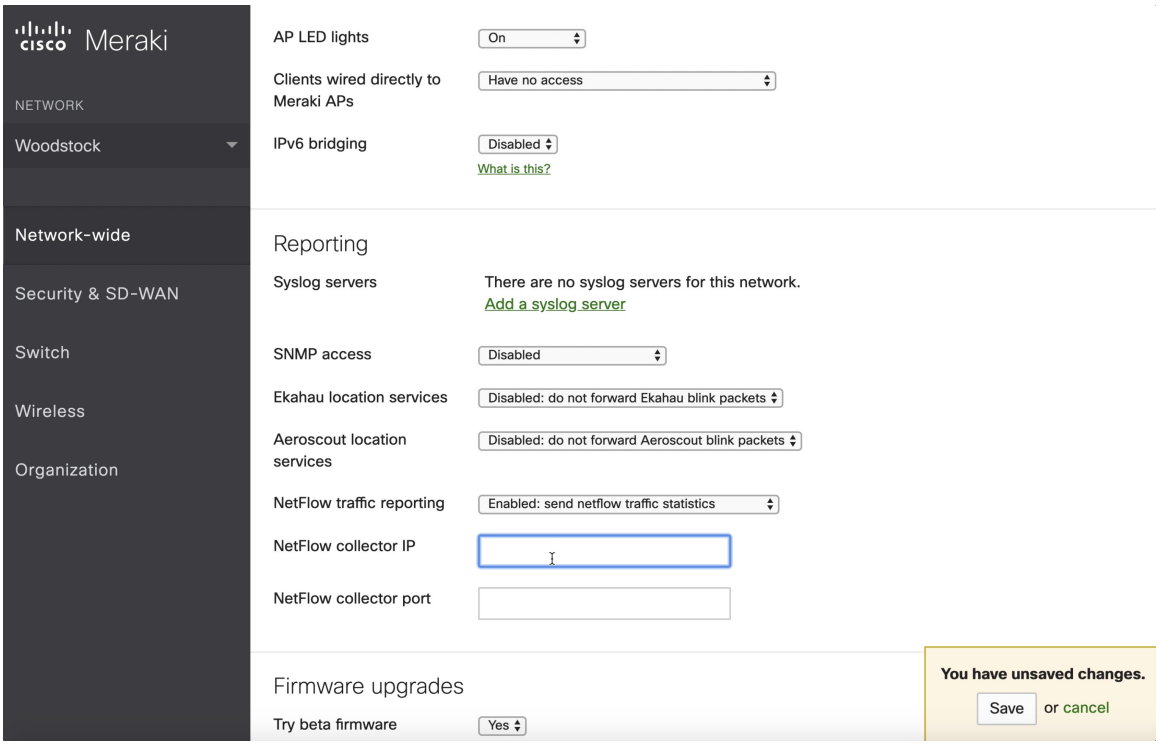
ステップ 1 Meraki UI コンソールにログインします。

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 *10 : 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、次の構成が許可されています。

- 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情報を提供することにより、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、構成が若干異なる場合があります。いずれの場合も、構成するシスコプラットフォームの公式シスコ コンフィグレーション ガイドも参照してください。

図 *11 : 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。

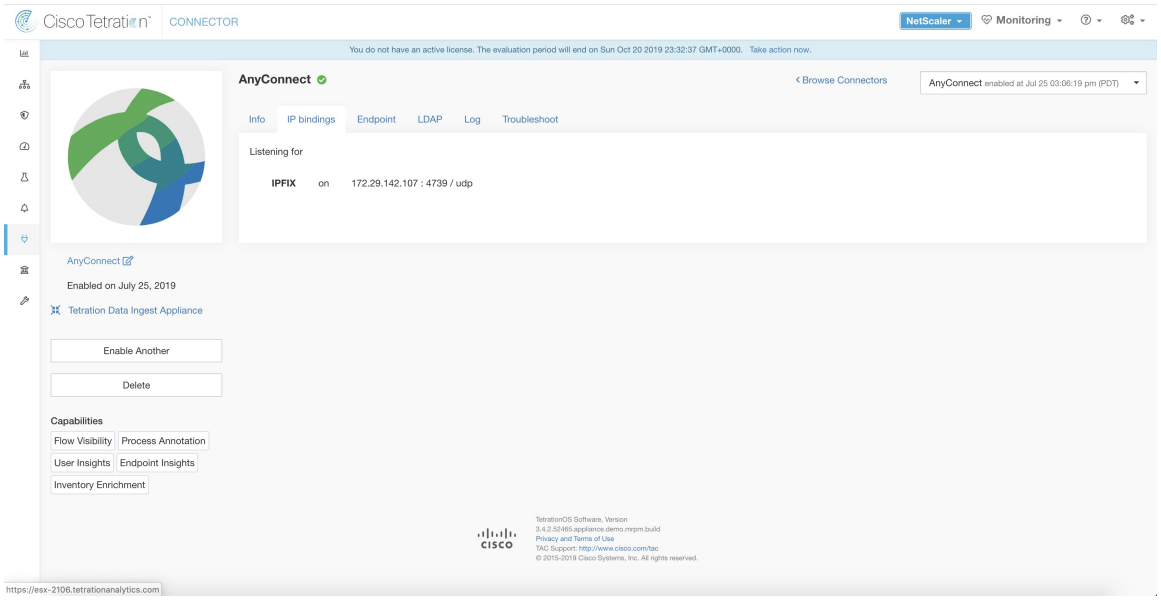
AnyConnect コネクタ

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 *12 : 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、次の設定が許可されています。

- *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情報を提供することにより、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

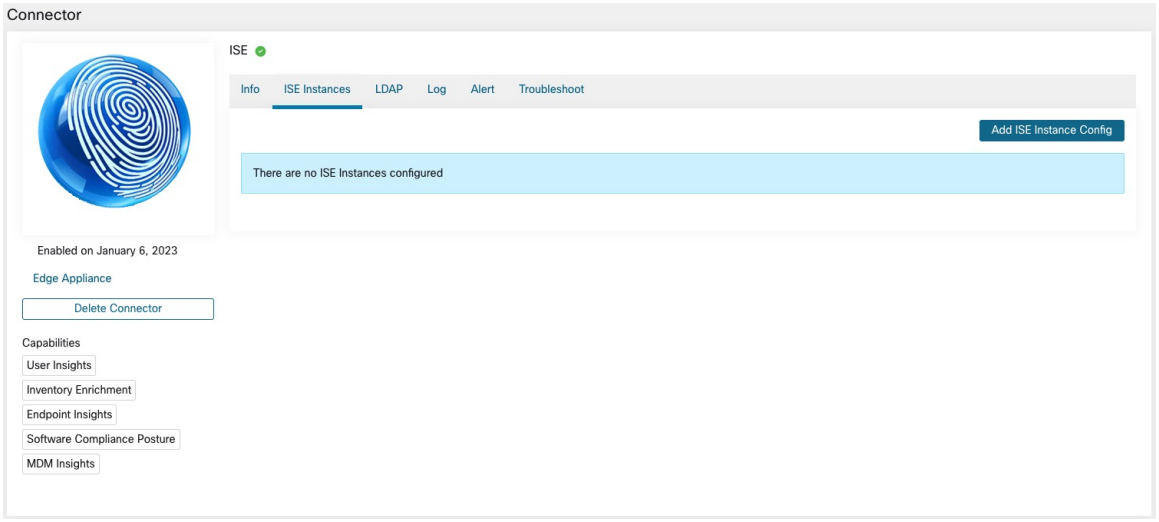
ISE コネクタ

ISE コネクタは、Cisco Platform Exchange Grid（pxGrid）を使用して Cisco Identity Services Engine に接続し、Cisco 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。

ISE コネクタは、次の高度な機能を実行します。

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 *13 : 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、[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、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+ が必要です。

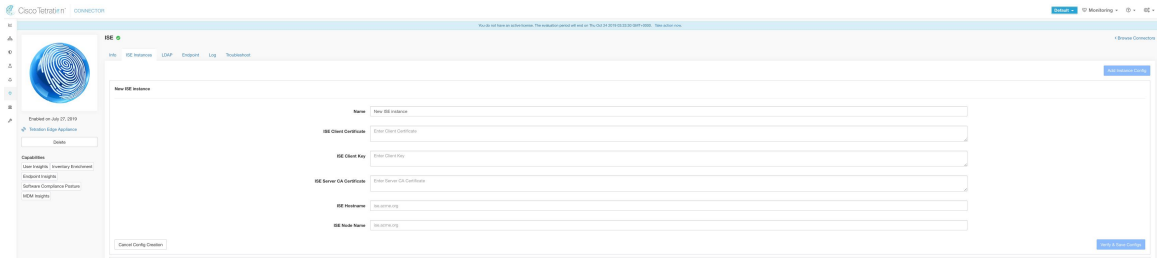
必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。ISE コネクタの場合、IPv4 および IPv6（デュアルスタックモード）アドレスがサポートされます。ただし、デュアルスタックのサポートはベータ機能であることに注意してください。

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 インスタンスの構成」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 *14 : 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。

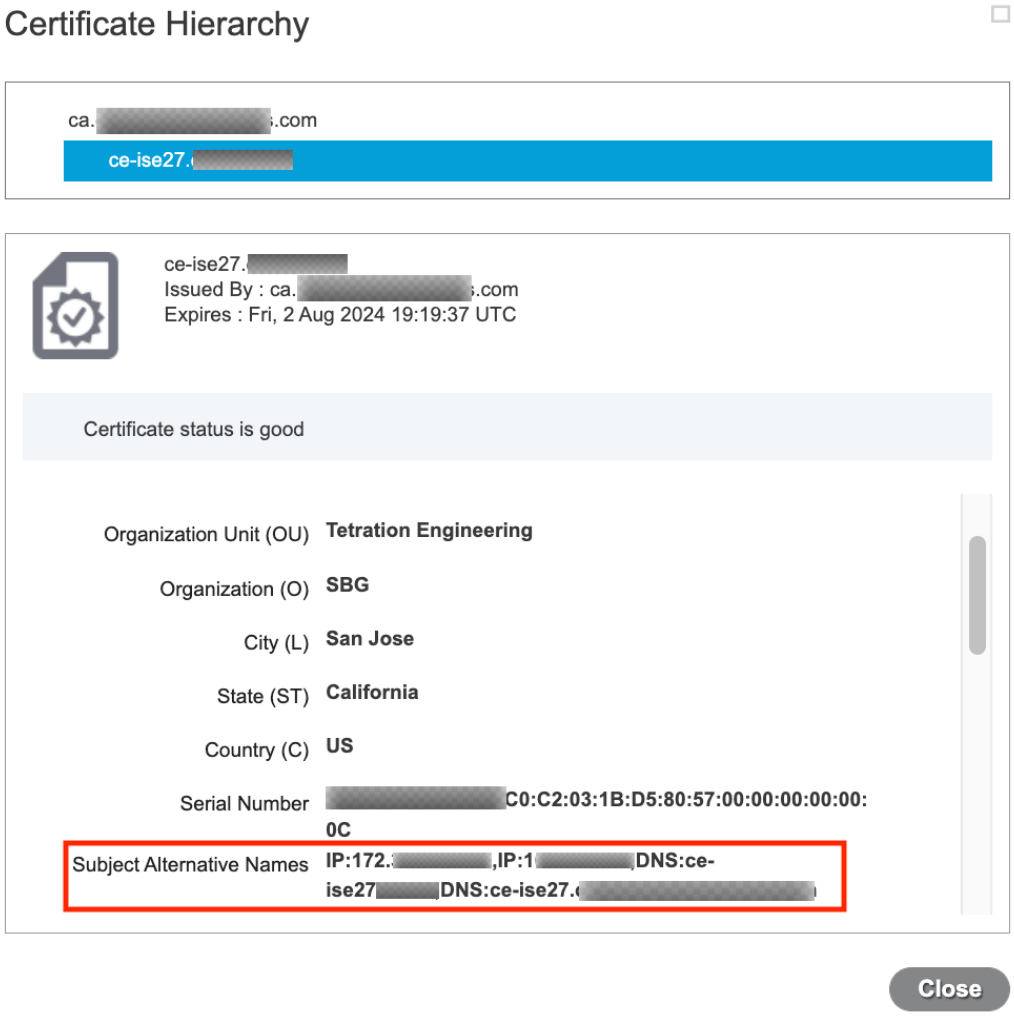
ステップ 1 **[管理（Administration）] > [システム（System）]** から [証明書（Certificates）] に移動します。

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 *15* : 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
```

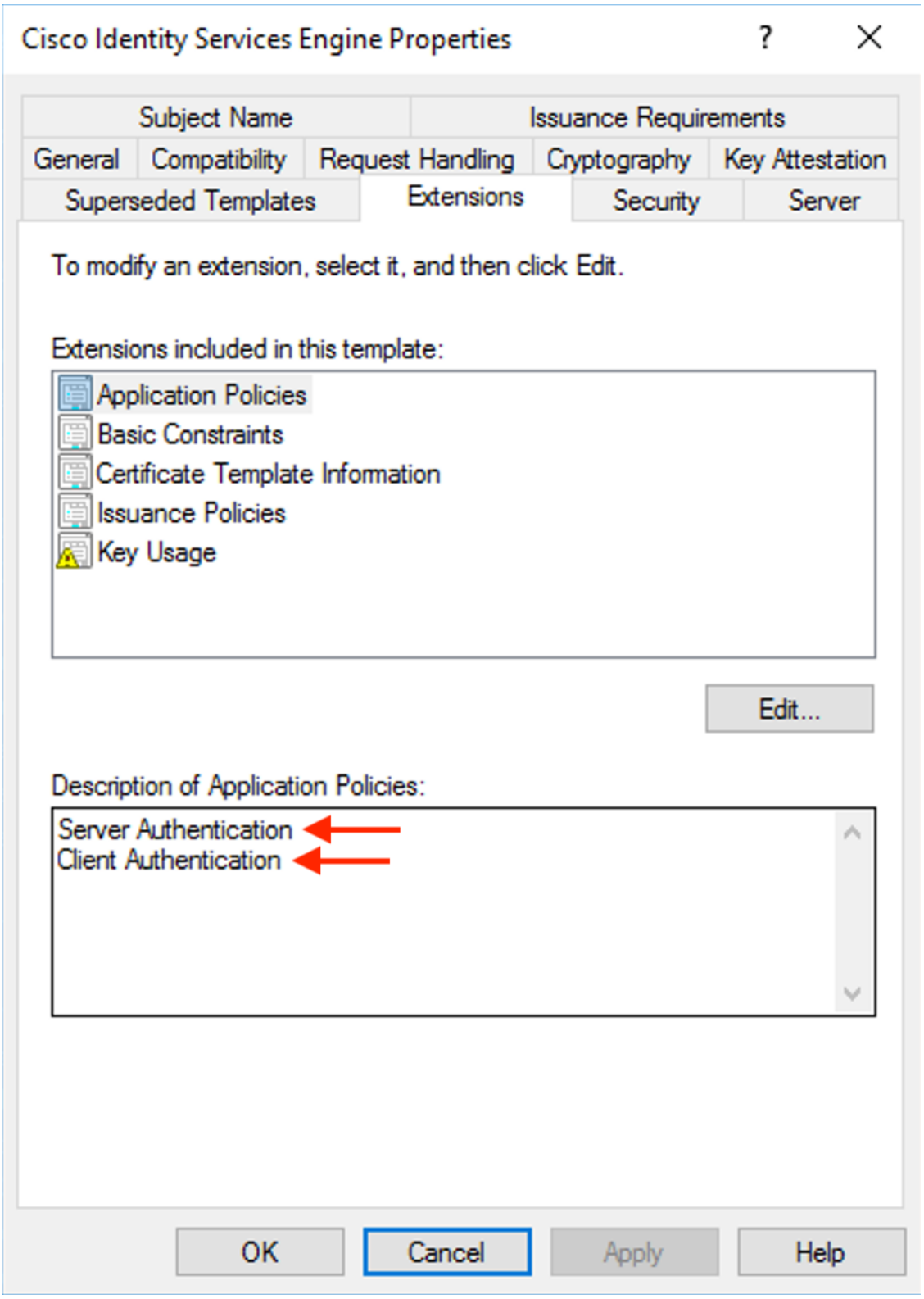
ステップ 6 Windows CA サーバーを使用して、CA によって証明書署名要求（CSR）に署名します。Windows CA サーバーも使用している場合は、次のコマンドを実行して pxGrid クライアントの CSR に署名します。

```
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 *16 :* 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


Cisco Identity Services Engine Properties
Subject Name
Issuance Requirements
General
Compatibility
Request Handling
Cryptography
Key Attestation
Superseded Templates
Extensions
Security
Server
To modify an extension, select it, and then click Edit.
Extensions included in this template:
Application Policies
Basic Constraints
Certificate Template Information
Issuance Policies
Key Usage
Edit...
Description of Application Policies:
Server Authentication
Client Authentication
OK
Cancel
Apply
Help

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
```

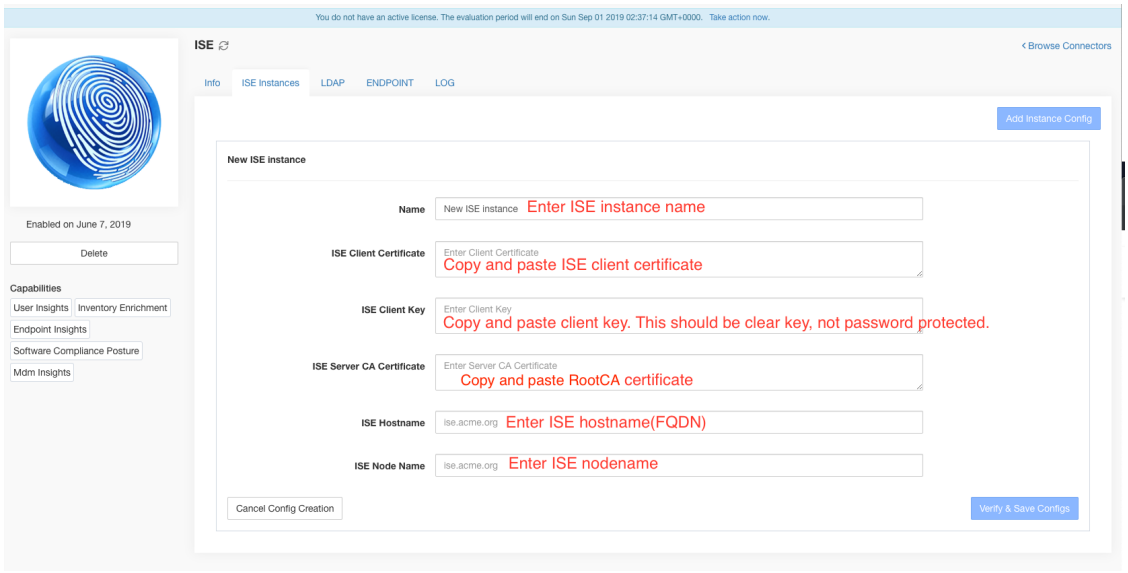
ステップ 8 次のコマンドを使用して、CA によって署名された PEM を確認することもできます。

```
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。

図 *17 : ISE* コネクタ設定



（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。

（注） 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

定期的に、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送信します。

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、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

（注） コネクタごとにサポートされる ISE エージェントの最大数は 400,000 です。

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

コネクタ	説明	仮想アプライアンス上に展開
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。

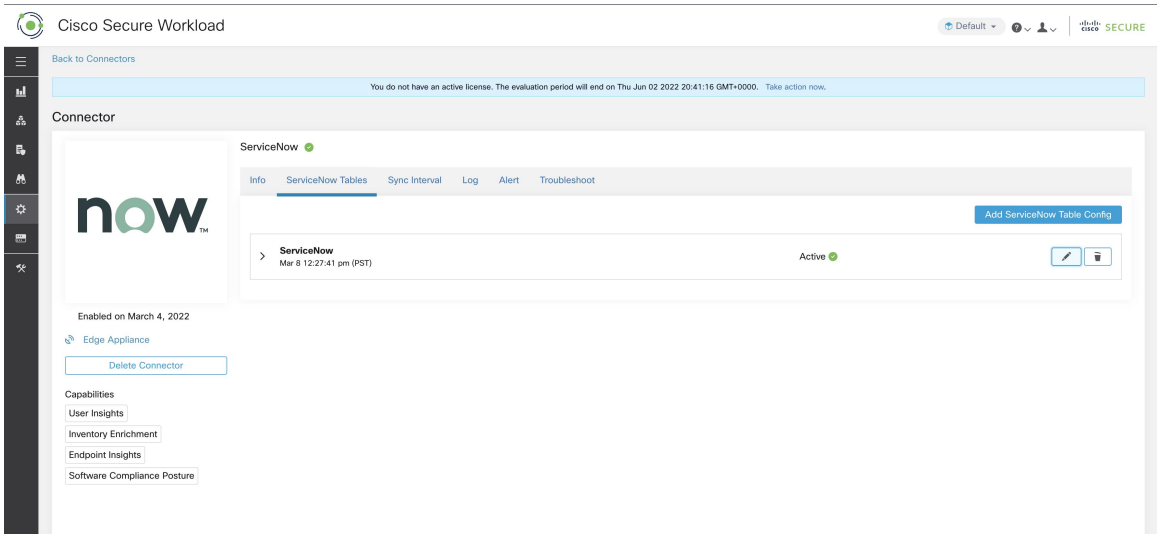
ServiceNow コネクタ

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 *18 : ServiceNow* コネクタ



ServiceNow コネクタの設定方法

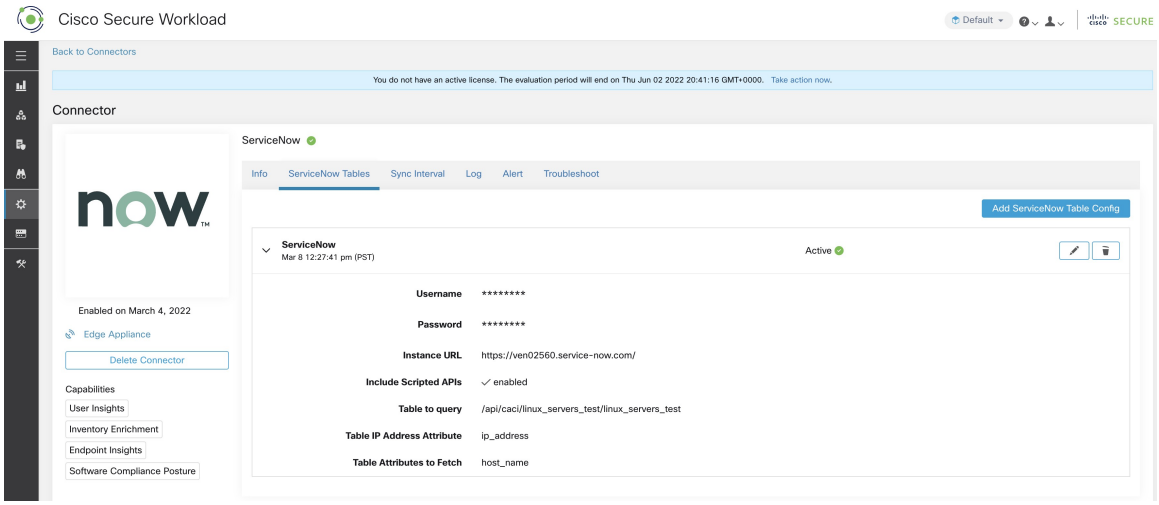
必要な仮想アプライアンスについては、「コネクタ用の仮想アプライアンス」を参照してください。コネクタでは、以下を設定できます。

- *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 *19 : 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 *20 : ServiceNow* インスタンス構成の最初のステップ

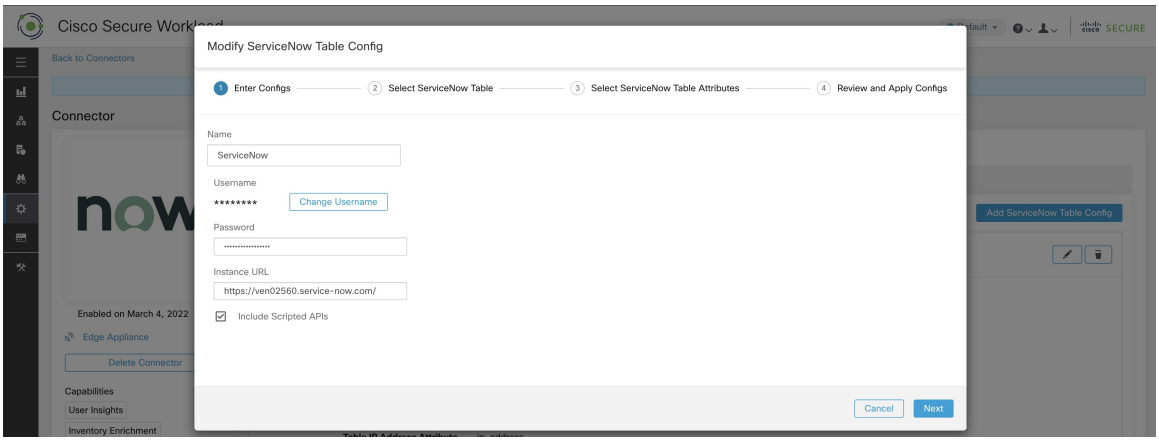


図 *21 : Secure Workload* が *ServiceNow* インスタンスからテーブル情報を取得します

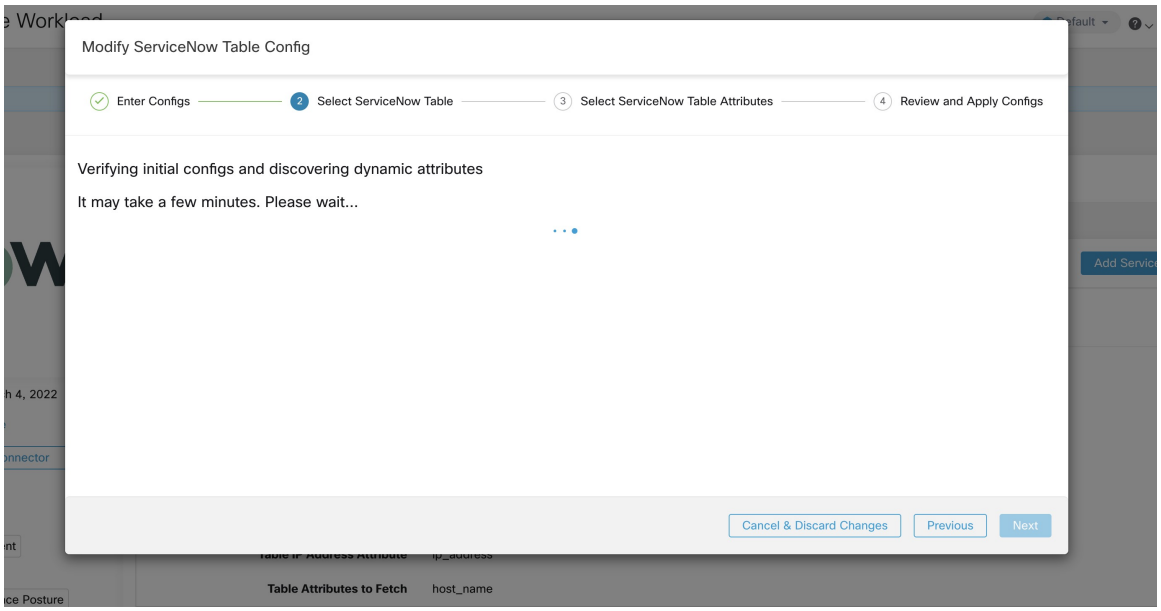


図 *22 : Secure Workload* がテーブルのリストを提示します

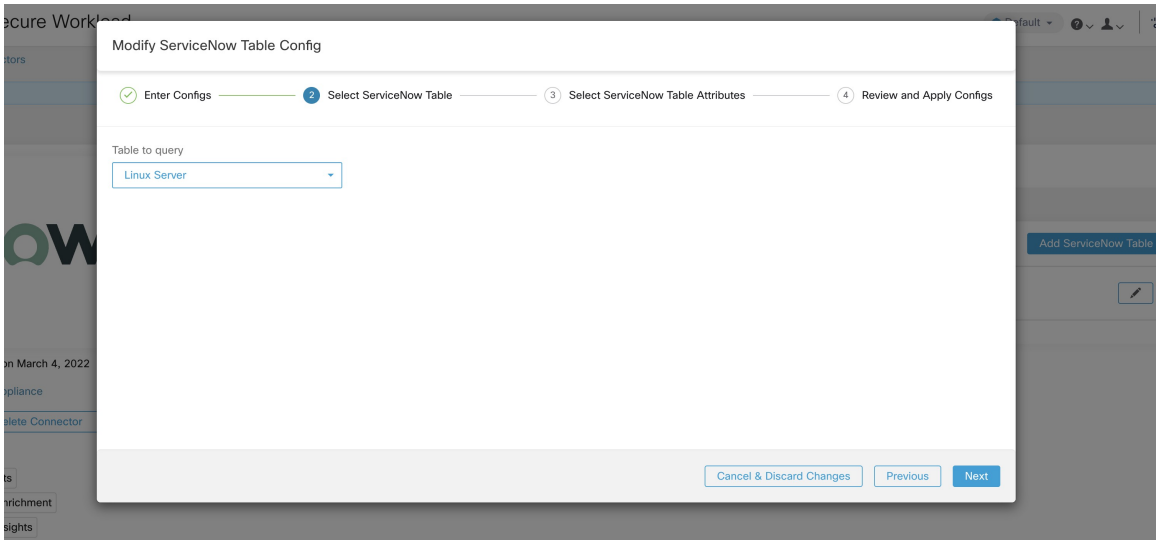


図 *23* : ユーザーが ***ip_address*** 属性とテーブル内の他の属性を選択します

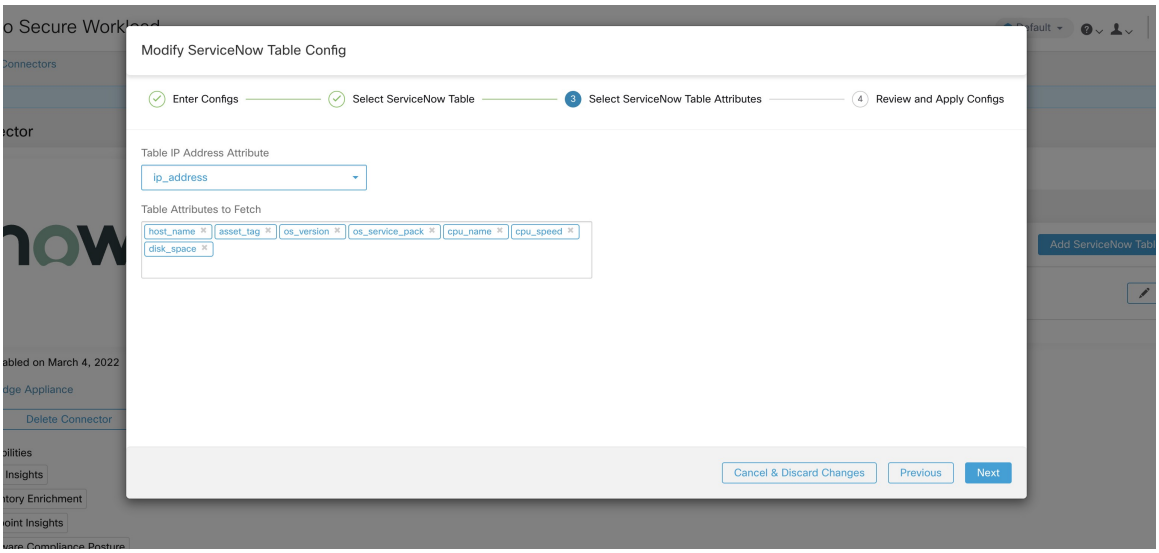
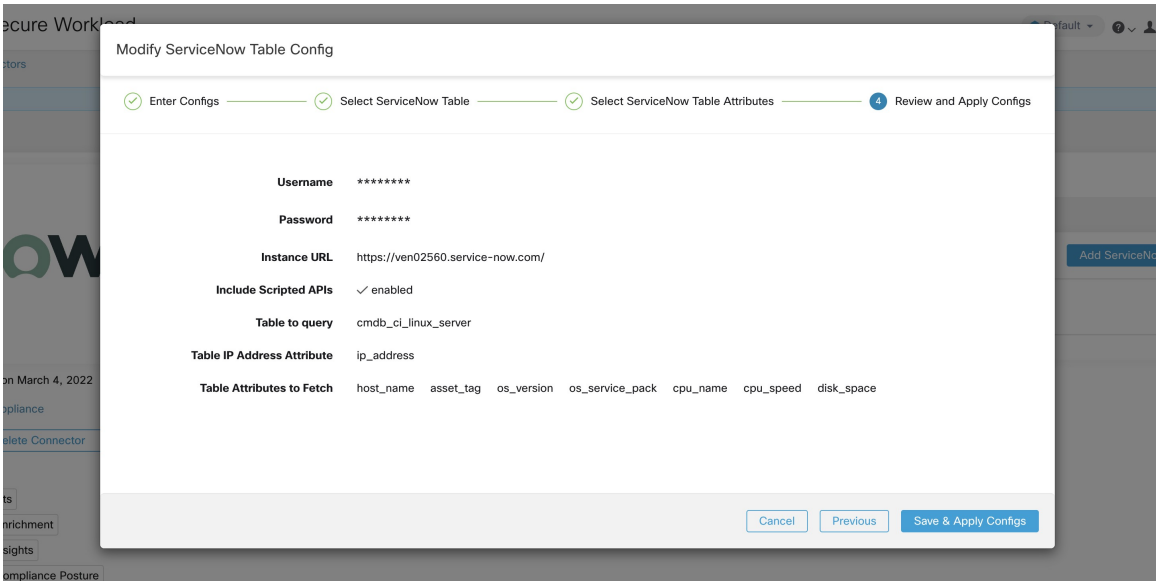


図 *24* : ユーザーが ***ServiceNow*** 構成を完了します



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。

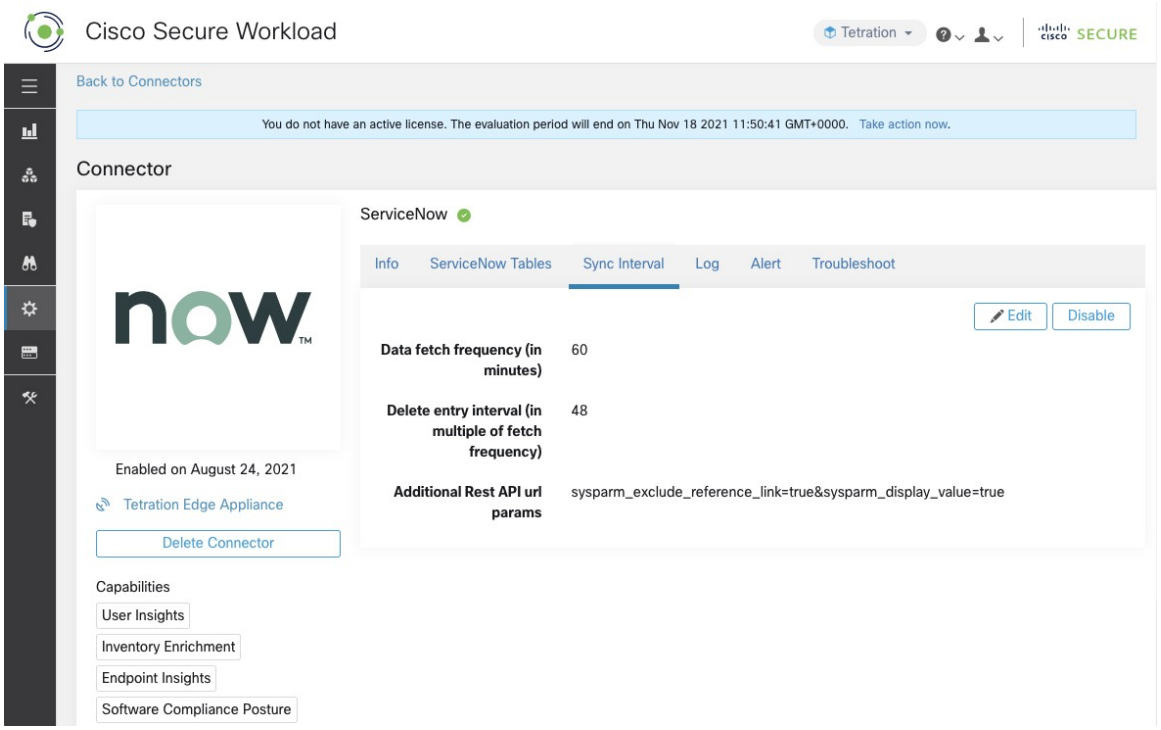
（注） Secure Workload は ServiceNow テーブルから定期的にレコードを取得します。これは、ServiceNow コネクタの [SyncInterval] タブで構成できます。デフォルトの同期間隔は 60 分です。エントリ数が多い ServiceNow テーブルと統合する場合は、この同期間隔をより高い値に設定する必要があります。

（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 *25 :* 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

TaaS クラスタの場合は、TaaS 運用チームに連絡して、ServiceNow ラベルのラベルをクリーンアップしてください。

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」をご確認ください。

[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。

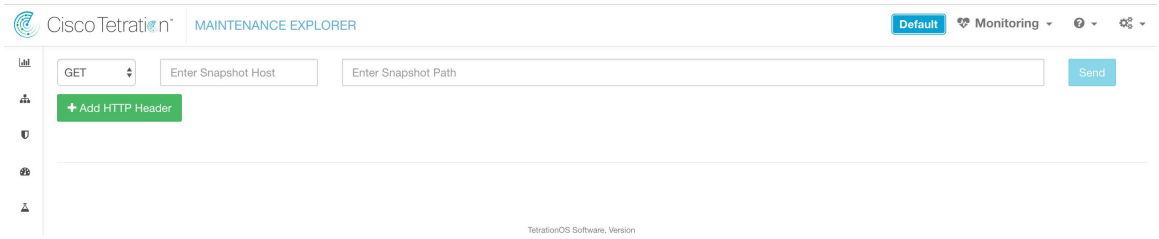
Explore コマンド UI へのアクセス

[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 *26 : [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

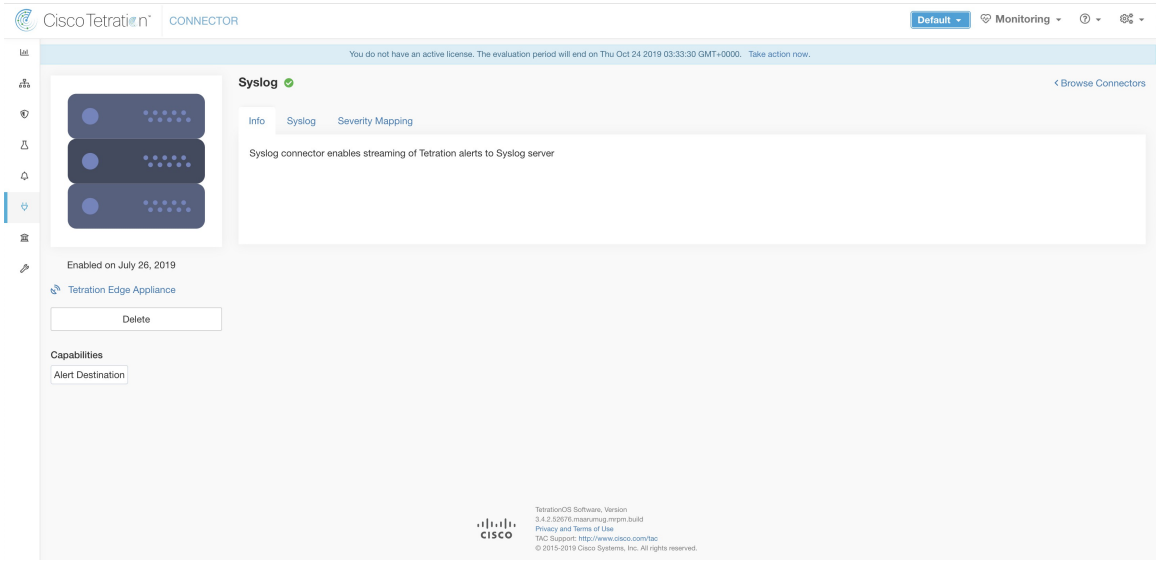
コネクタ	説明	仮想アプライアンス上に展開
Kinesis	Amazon Kinesis で Secure Workload アラートを送信します。	Secure Workload Edge

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 *27 : 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 *28 : Syslog Connector* の設定例

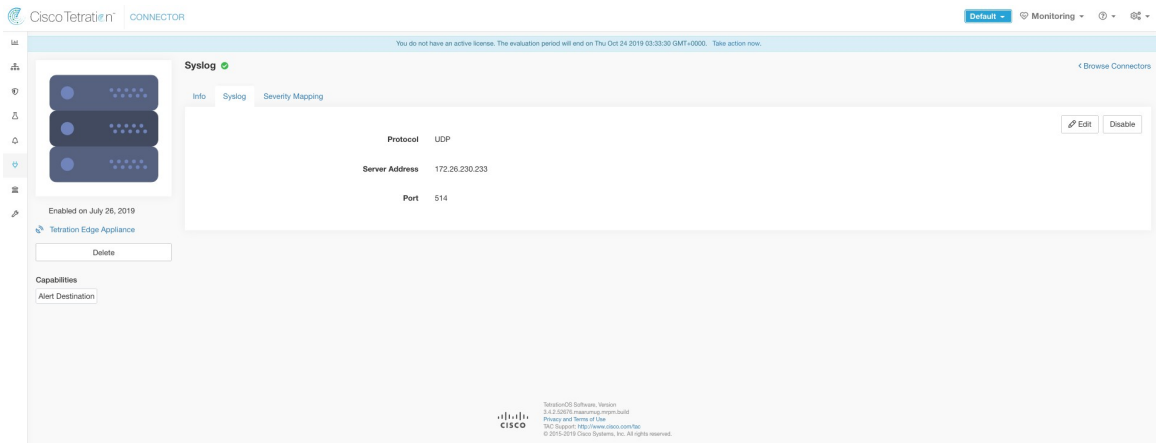


図 *29 :* サンプルアラート

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。

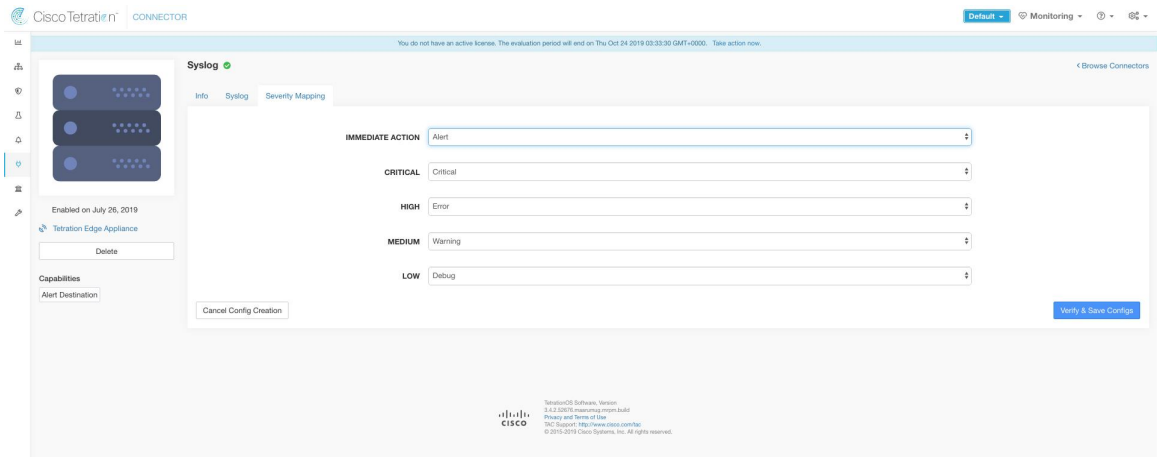
安全なワークロードアラートのシビラティ（重大度）	Syslog のシビラティ（重大度）
LOW	LOG_DEBUG
[中（Medium）]	LOG_WARNING
HIGH	LOG_ERR

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）
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 ***30 : Syslog*** のシビラティ（重大度）のマッピング例



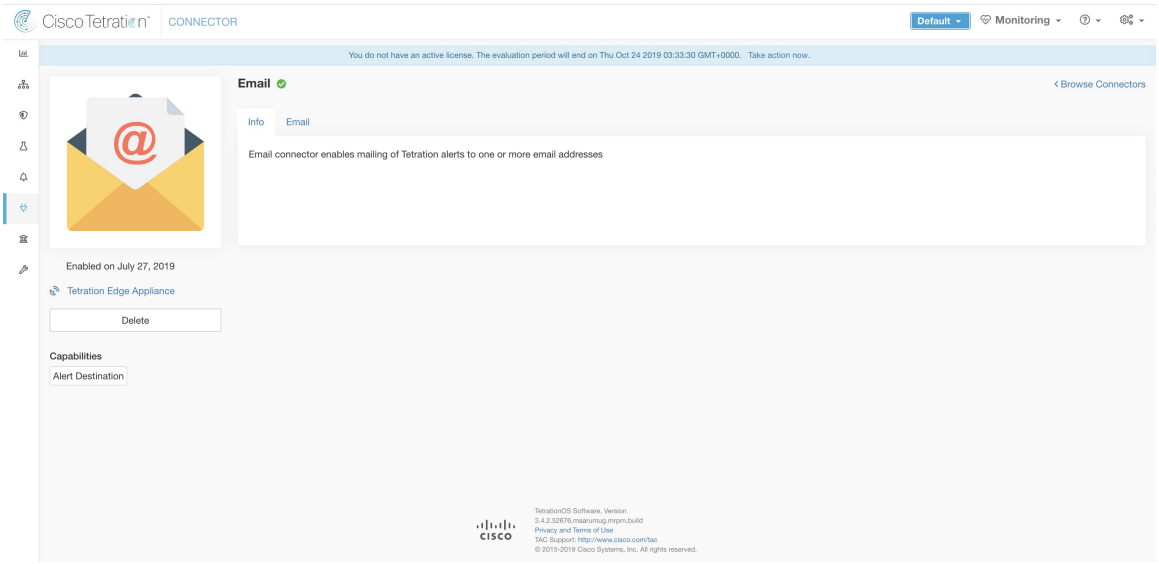
制限

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 *31 : 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 *2 :* 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。

パラメータ名	タイプ	説明
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	電子メールアドレスのカンマ区切りリスト。

図 ***32 : Email Connector*** の設定例

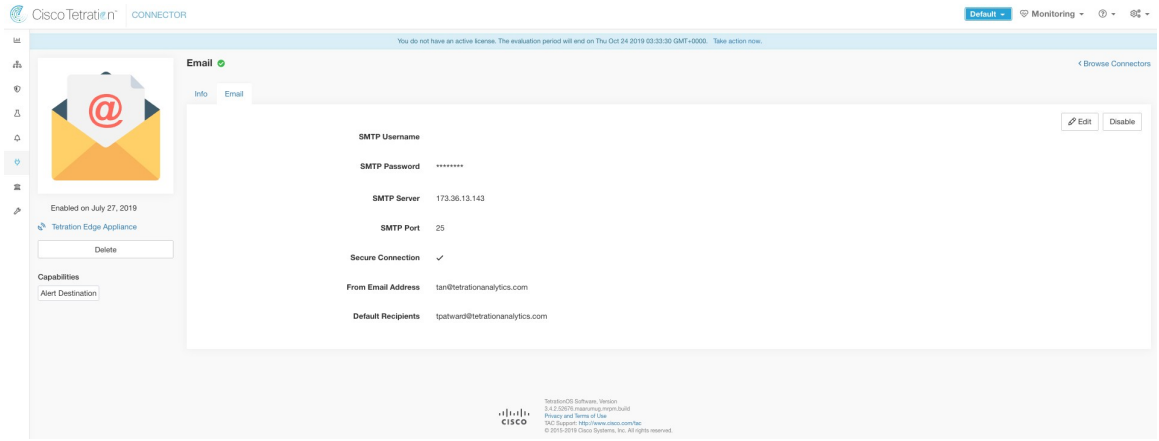
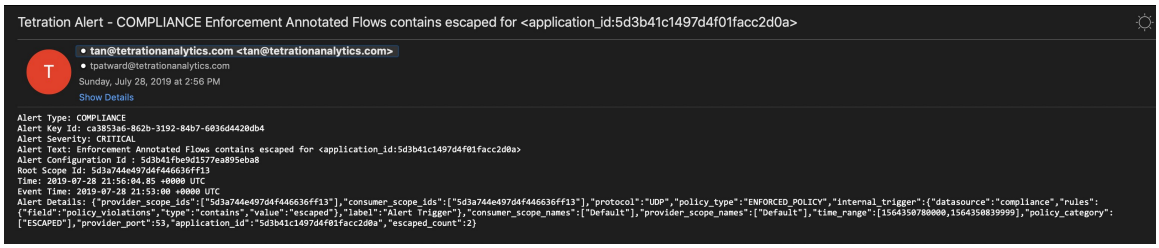


図 ***33 :*** サンプルアラート



（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。

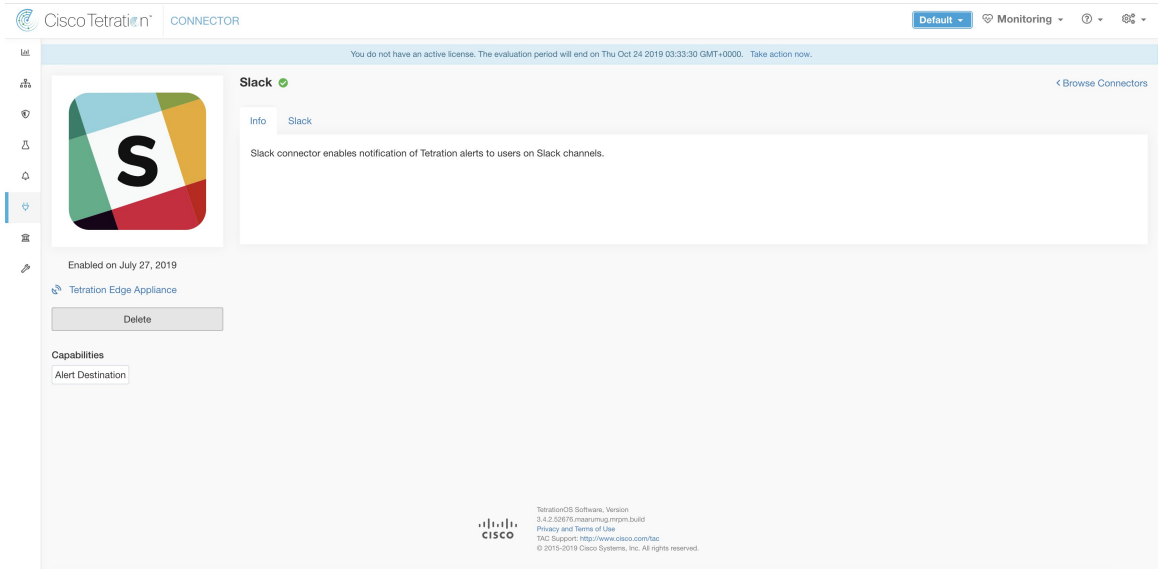
制限

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
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 *34 : 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 *35 : Slack* コネクタの設定例

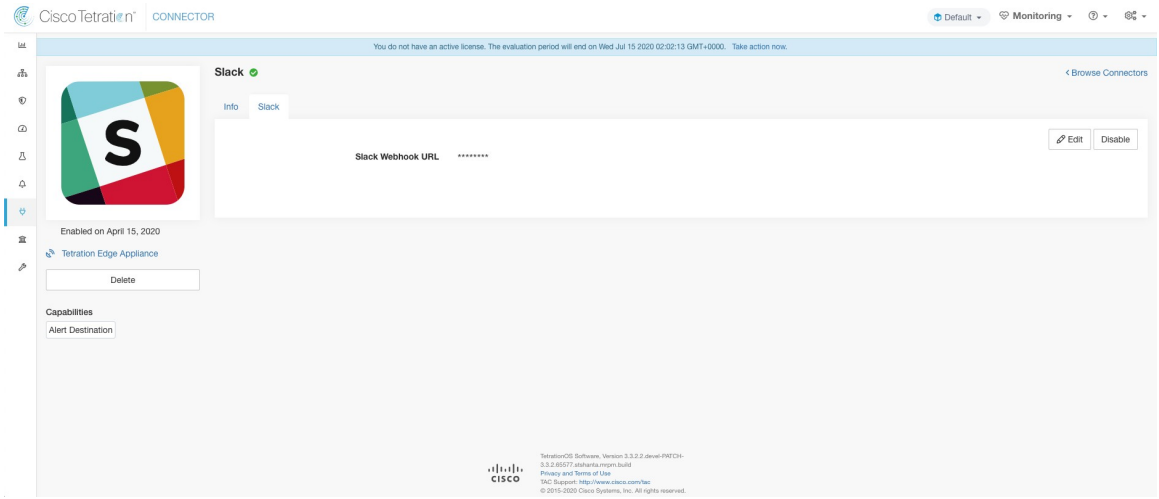
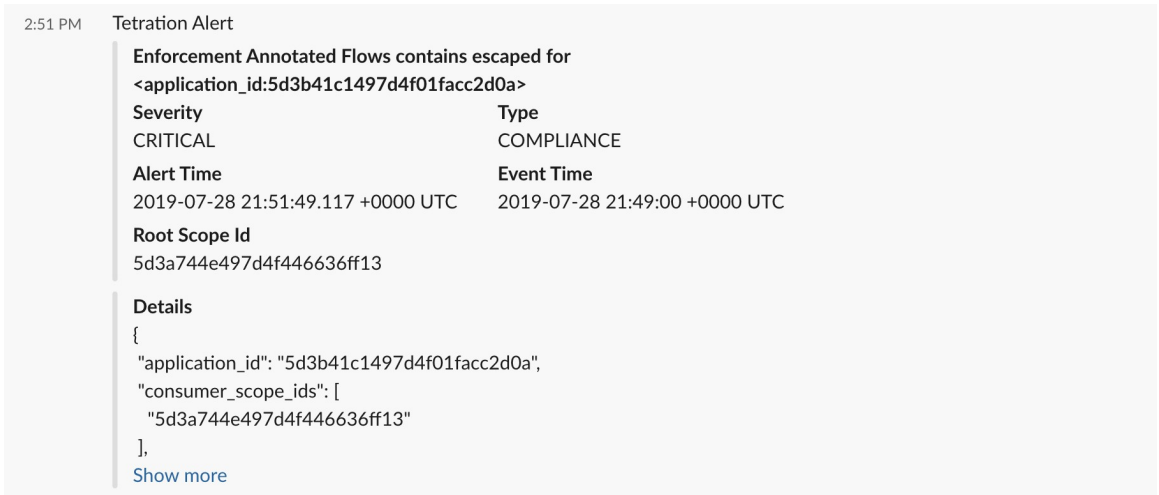


図 *36 :* サンプルアラート



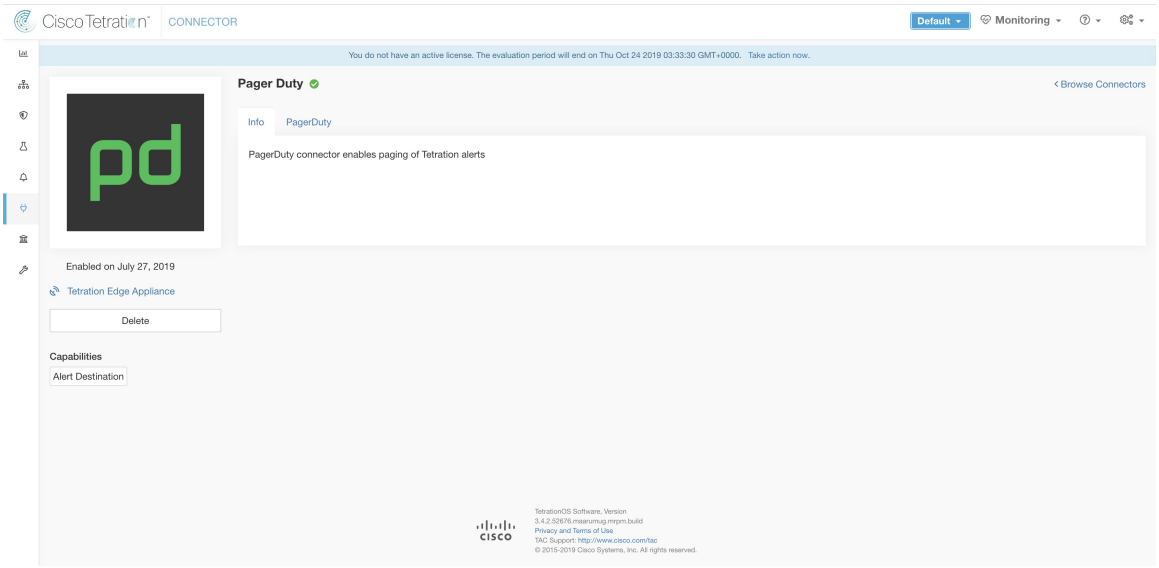
制限

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 *37 : 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 *38 : PagerDuty* コネクタの構成例

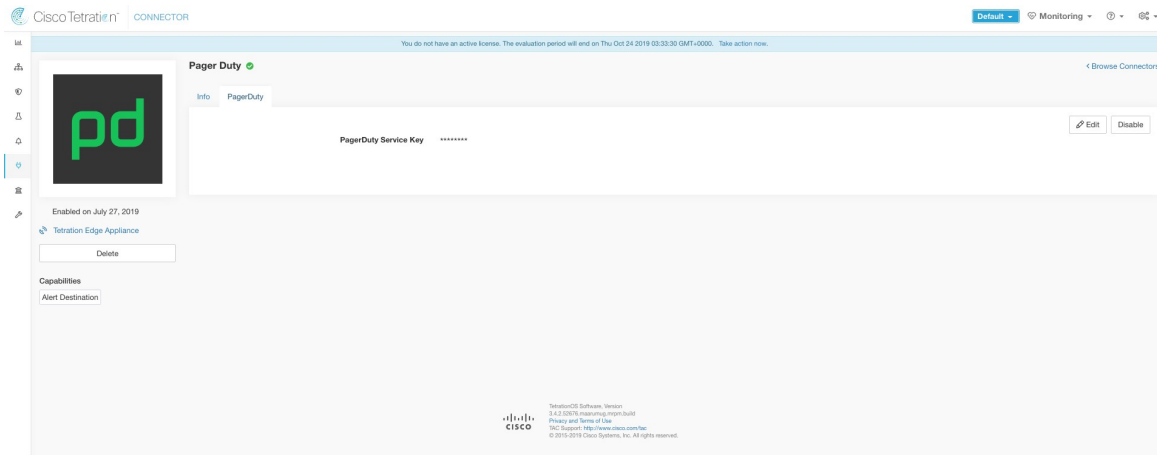
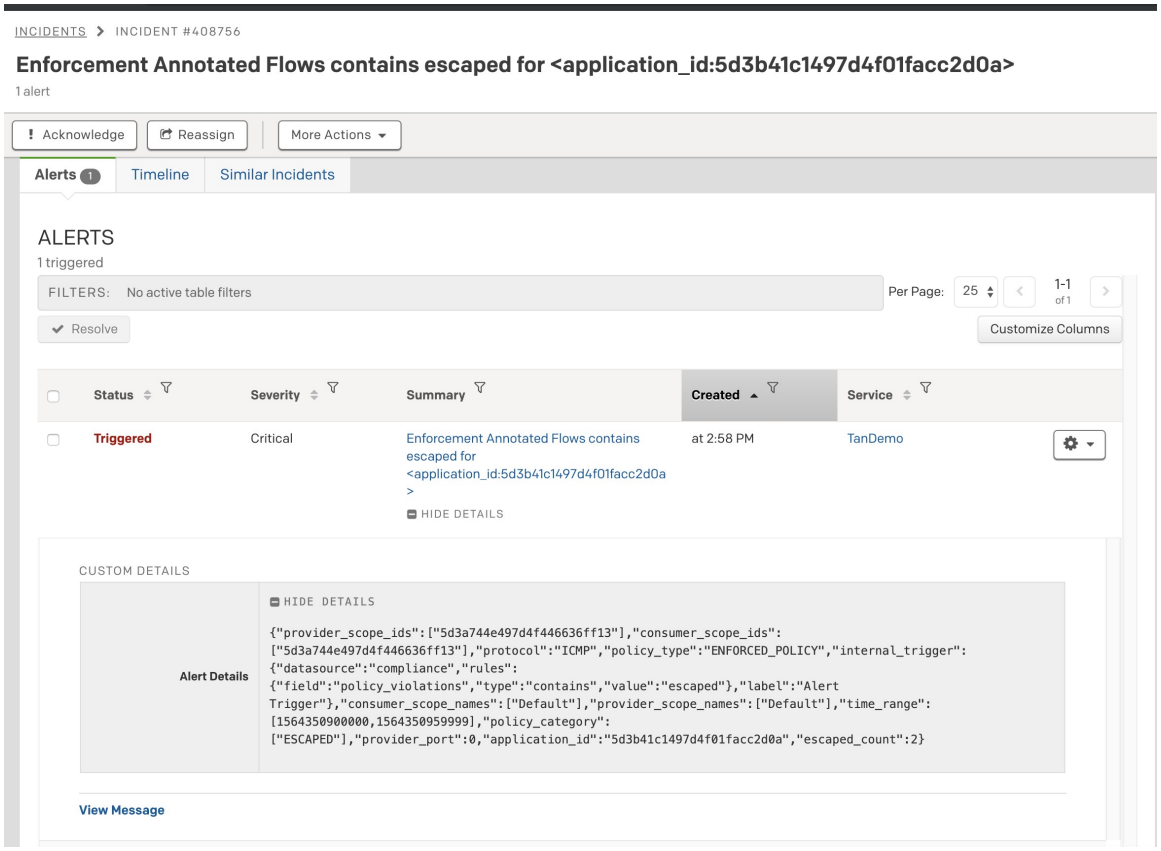


図 *39 :* サンプルアラート



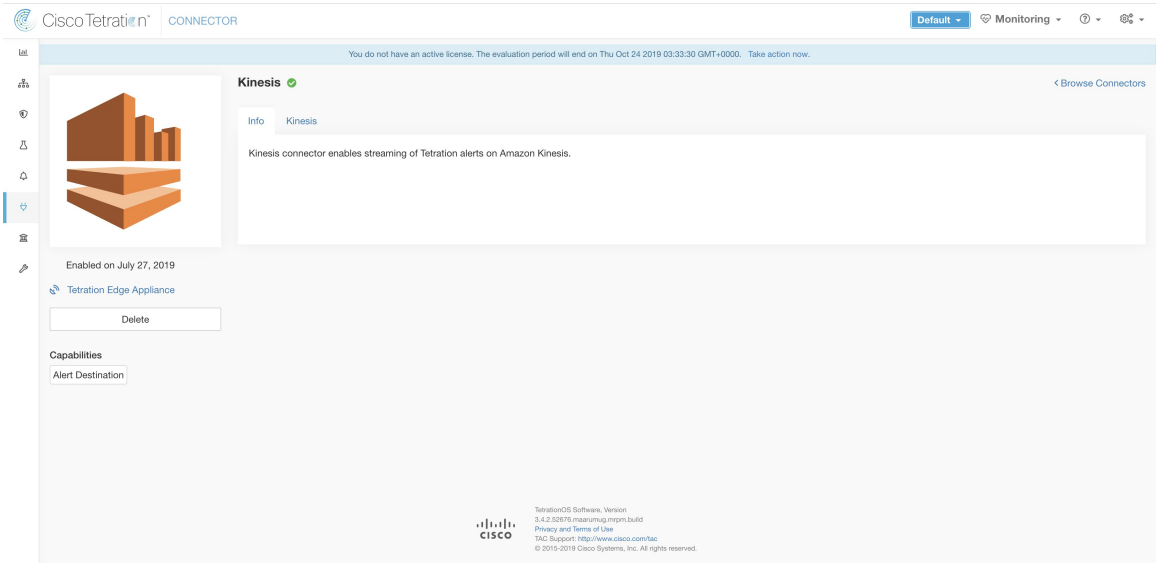
制限

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 *40 : Kinesis* コネクタ

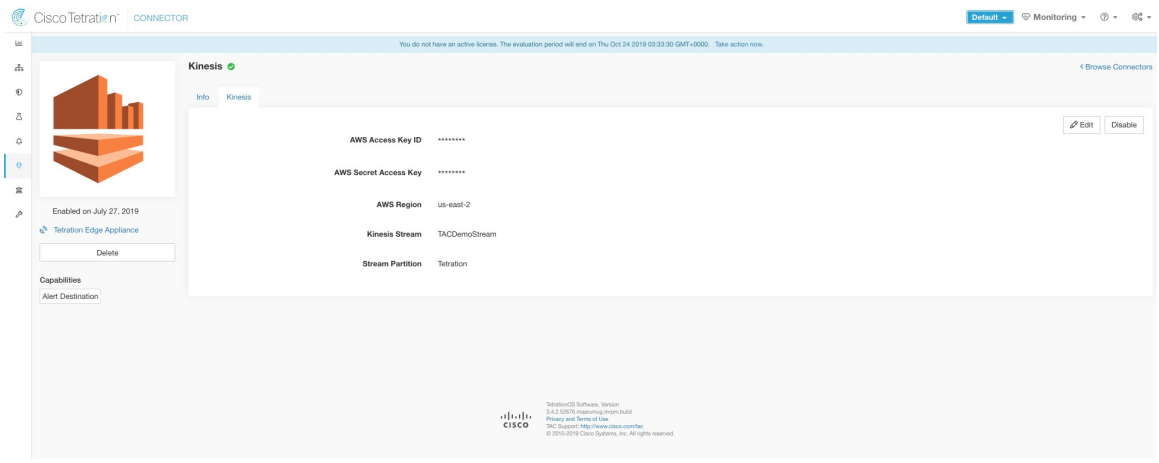


次の表は、Amazon Kinesis で Secure Workload アラートを公開するための設定の詳細を示しています。詳細については、「Kinesis 通知設定」を参照してください。

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

パラメータ名	タイプ	説明
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 *41 : 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 Secure Workload 機能を使用するには、Cloud Connector を使用します。

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 EKS Kubernetes 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 AKS Kubernetes クラスタから： • メタデータの収集	該当なし
GCP	GKE Kubernetes 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） AWS 中国リージョン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。

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 AWS 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」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。

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「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。

   管理ポリシー JSON の例：

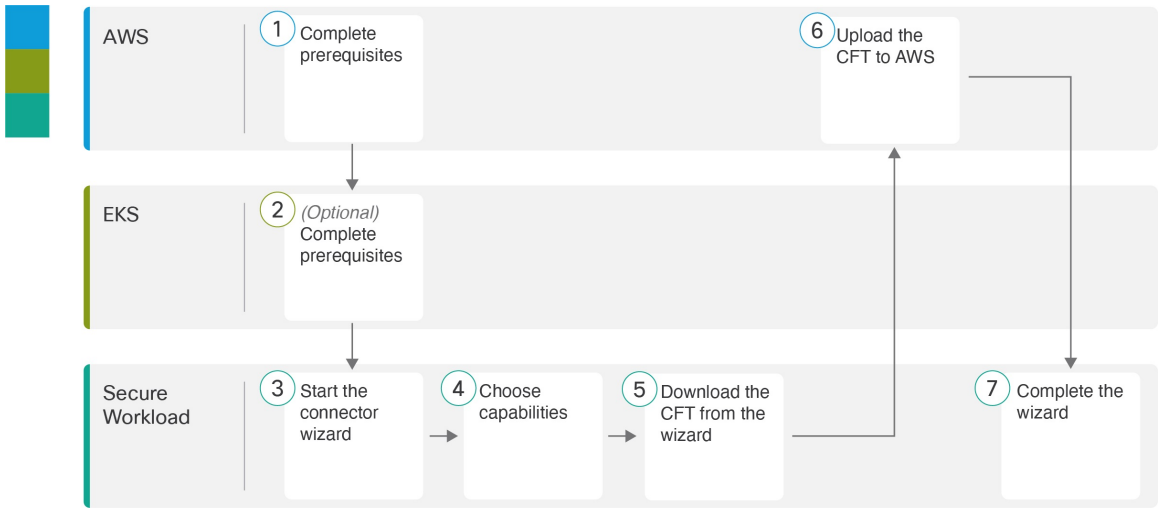
```
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "VisualEditor0",
"Effect": "Allow",
"Action": "sts:AssumeRole",
"Resource": [
<AWS_role_cross_account_1_arn>, <AWS_role_cross_account_2_arn>...
]
}
]
}
```

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」を参照してください。

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」）を参照してください。

図 *42 : 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。）

AWS コネクタの設定

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、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 （80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。

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 [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。

- Cloud Connector 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」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。

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） 中国リージョン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

すべての機能について：コネクタに関連するすべての設定は、複数のサブスクリプションに属している場合があります。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。

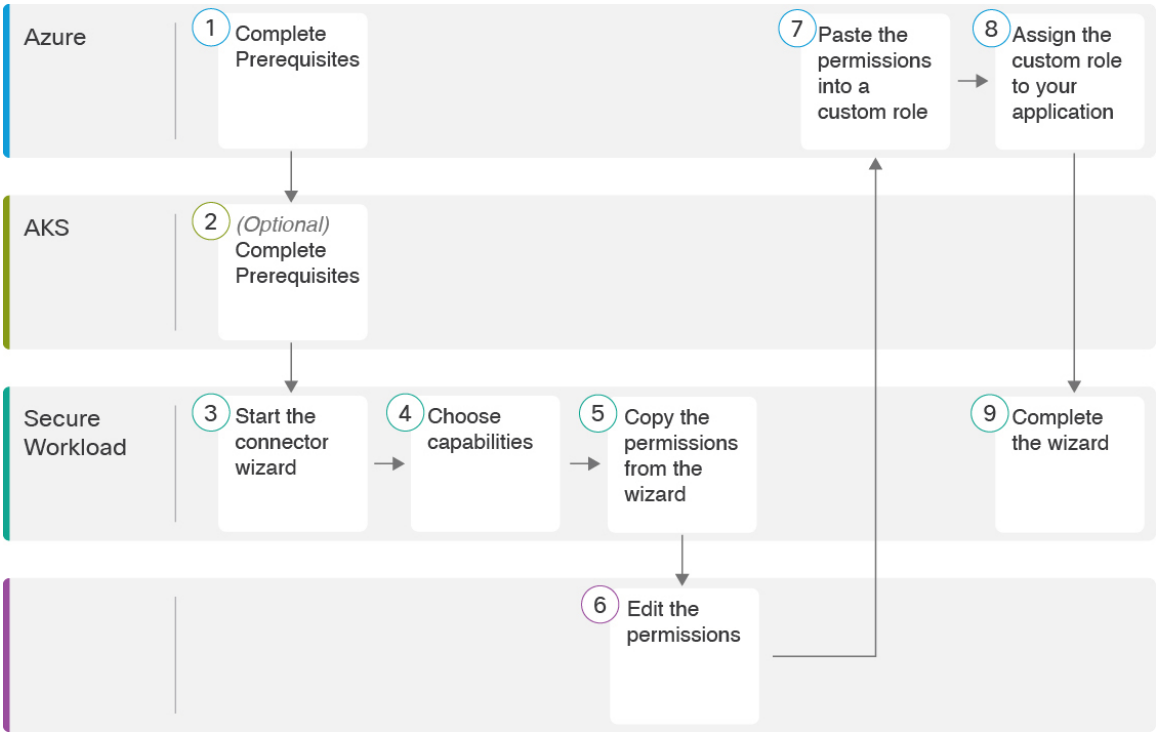
以下のAzure Inventory にセグメンテーションポリシーを適用するときのベストプラクティス（90 ページ）も参照してください。

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 *43 : 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

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。
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 （91 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**。
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 および各 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 および 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 （90 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 （90 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「Kubernetes/Openshift エージェント：優れた可視性と適用」を参照してください。

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。

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 （86 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 （90 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 （89 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。

- Cloud Connector 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。

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

GCP コネクタの設定

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 「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。

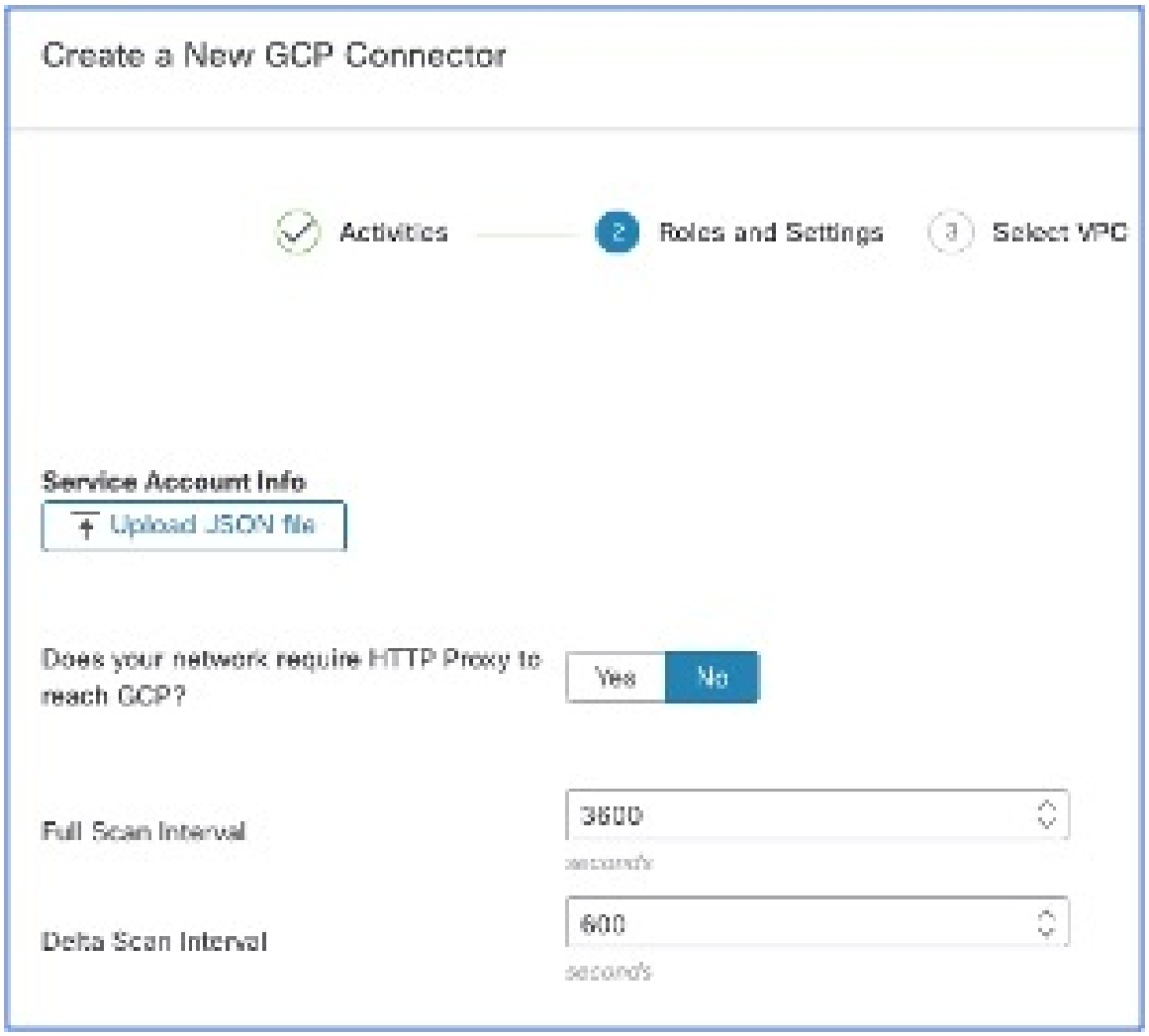
ステップ 5 コネクタに名前を付けた後、[次へ（Next）] をクリックします。

ステップ 6 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。

ステップ 7 設定を次のように構成します。

属性	説明
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。

図 *44 : GCP* コネクタ設定の概要



ステップ 8 [次へ（Next）] をクリックします。システムが GCP サービスアカウントから 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 9 GKE クラスタのリストから、メタデータを収集するクラスタを選択します。

親仮想プライベートクラウド（VPC）を選択し、必要に応じて GKE クラスタを有効にします。

ステップ 10 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 11 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 12 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。

次の手順：

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」を参照してください。

GCP コネクタの編集

異なる GKE クラスタや追加の 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 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。

選択した VPC 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。

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。コネクタによって取り込まれたデータを削除しても、そのコネクタは削除されません。

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（Connector）]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

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。

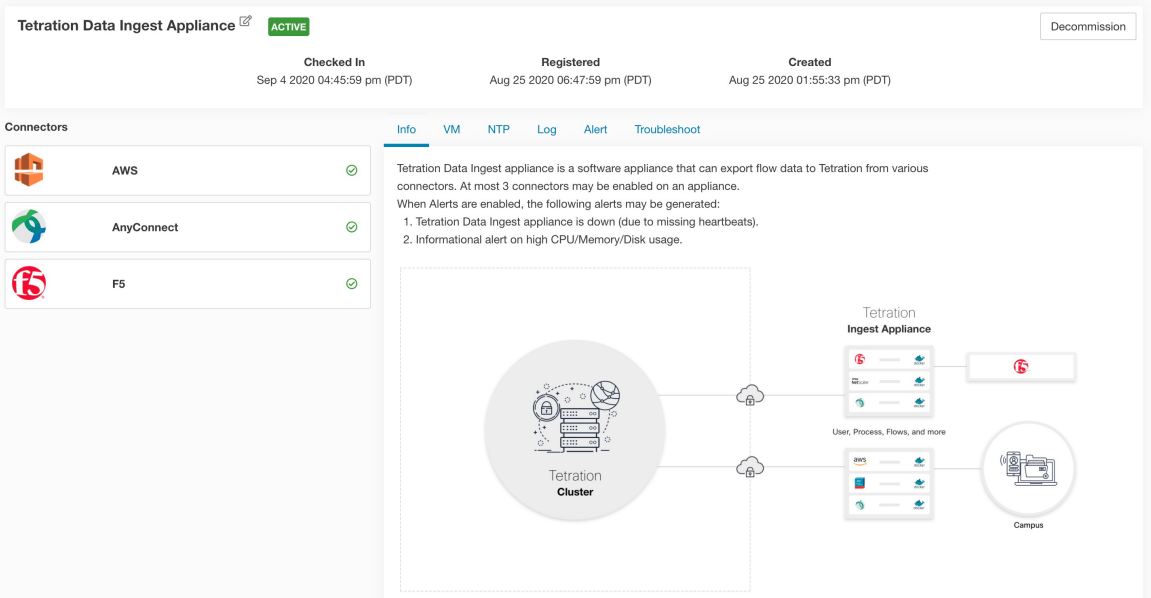
仕様

- 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

「コネクタ用の Cisco Secure Workload 仮想アプライアンス」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 *45 : 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 *46 : 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。

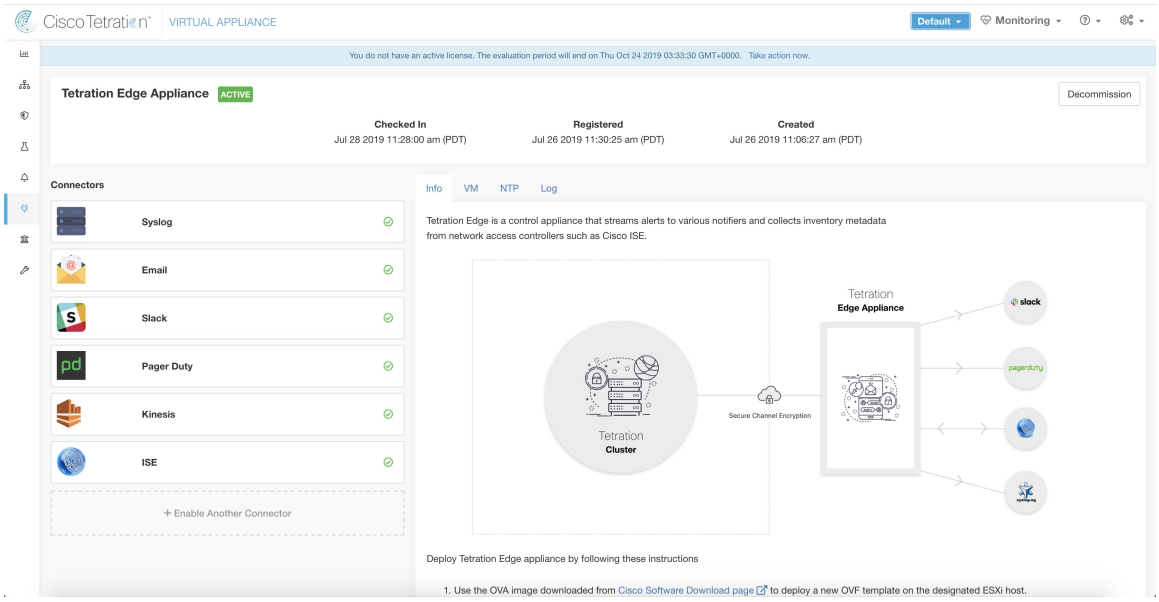
仕様

- 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

「コネクタ用の Cisco Secure Workload 仮想アプライアンス」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 *47 : 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。

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。

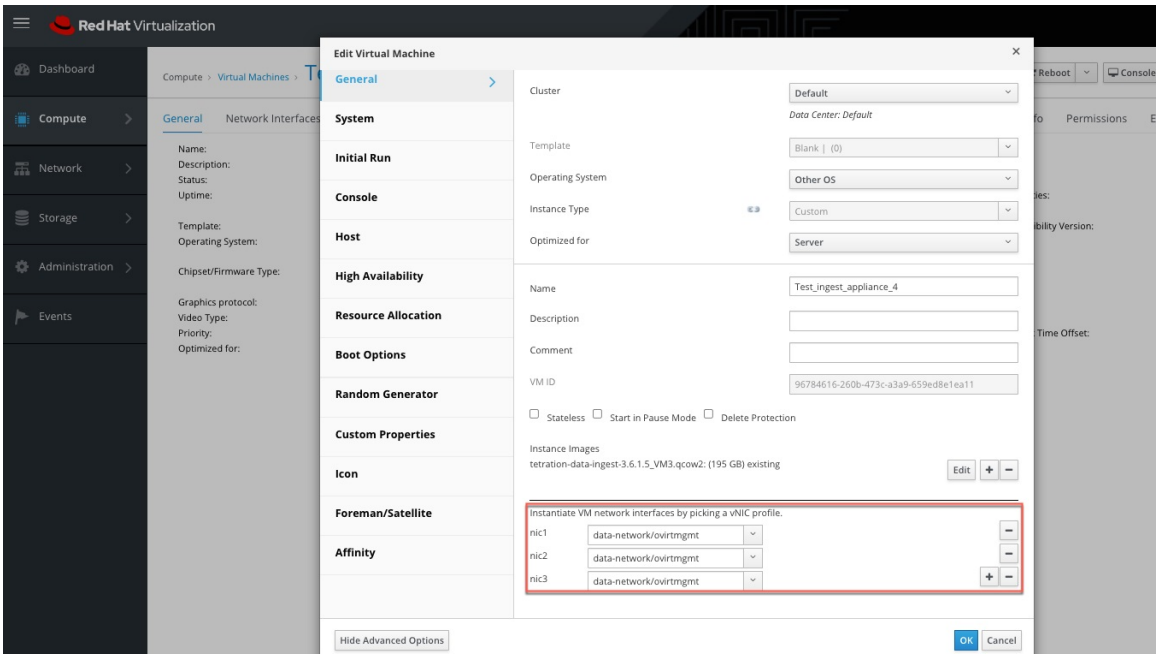
ステップ 12 次のステップで、VM 設定バンドルが生成され、ダウンロードできるようになります。VM 設定バンドルをダウンロードします。「*VM* 設定バンドルのダウンロード」のスクリーンショットを参照してください。

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。

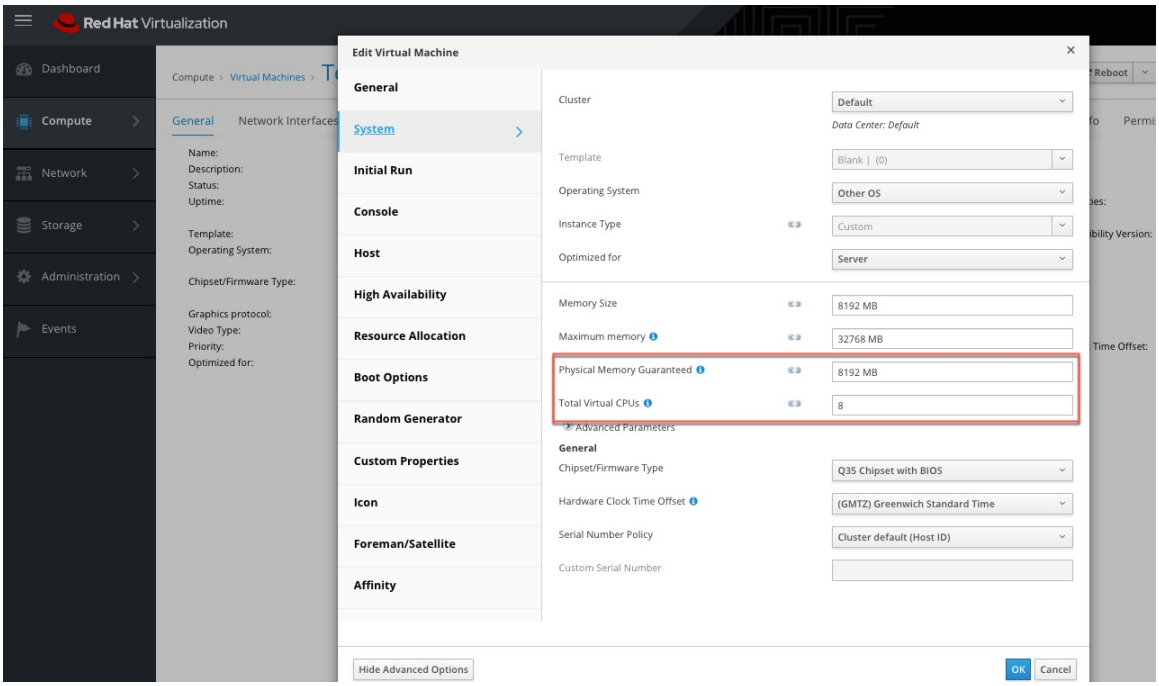
- Ingest アプライアンスの場合、3 つのネットワーク インターフェイスを設定します。

図 *48 : 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 *49 : 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 *50 : Secure Workload Ingest* アプライアンスの展開

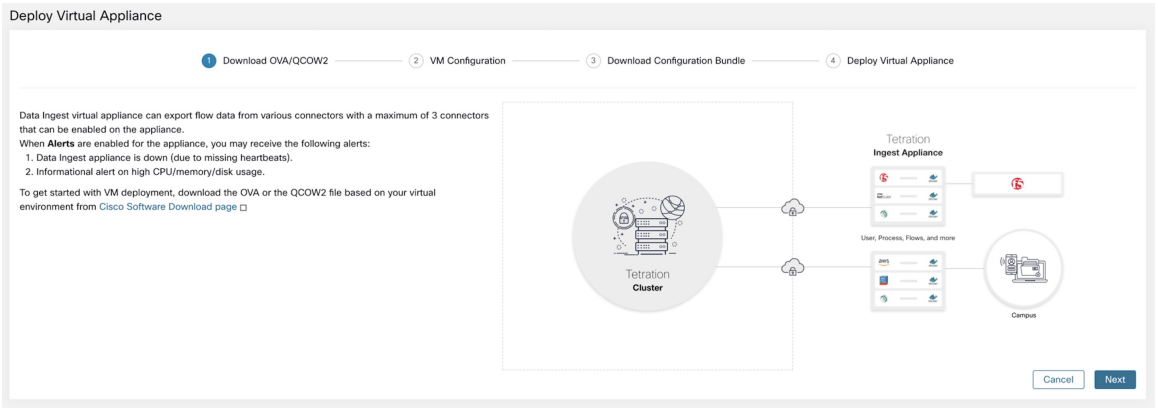


図 *51* : ネットワークパラメータを使用した *VM* の設定

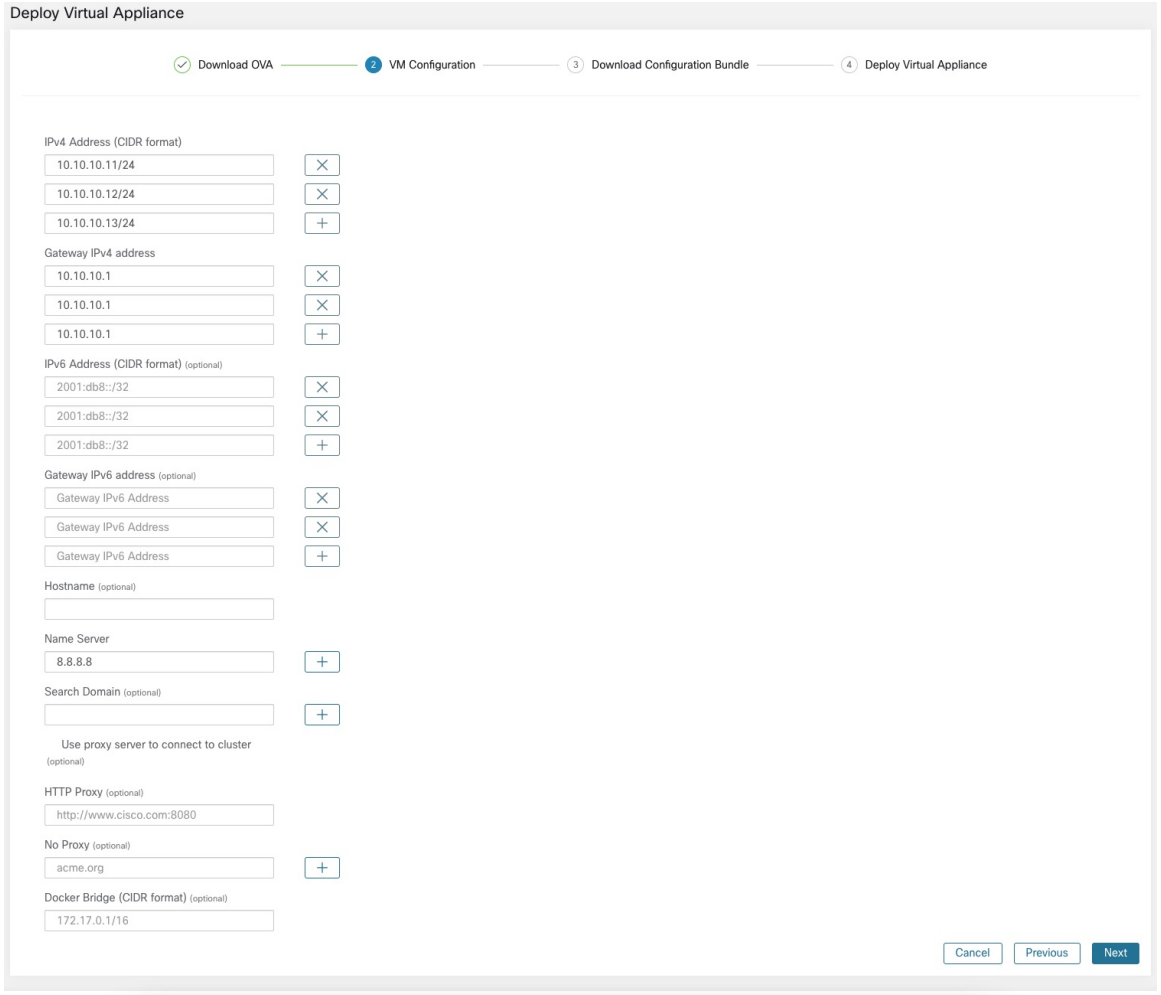


図 *52* : *VM* 設定バンドルのダウンロード

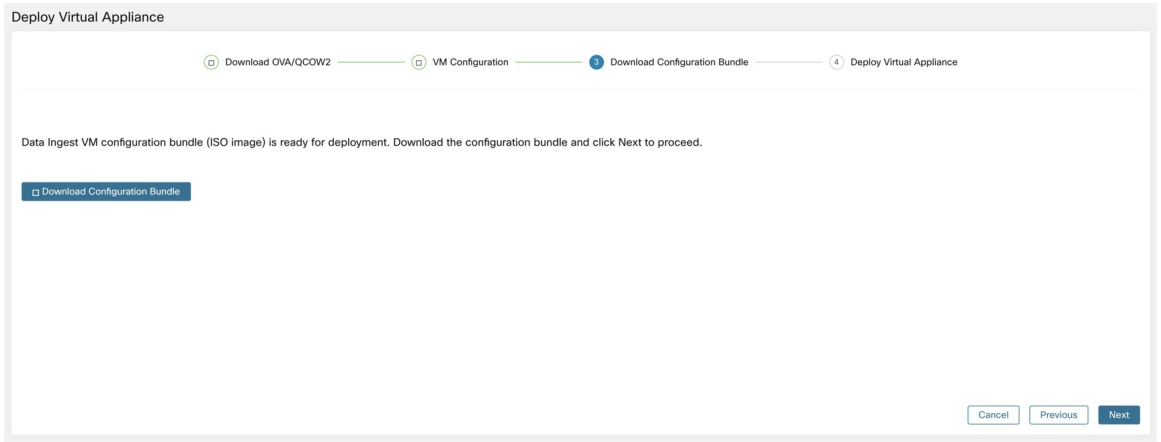


図 *53 : VM* の展開

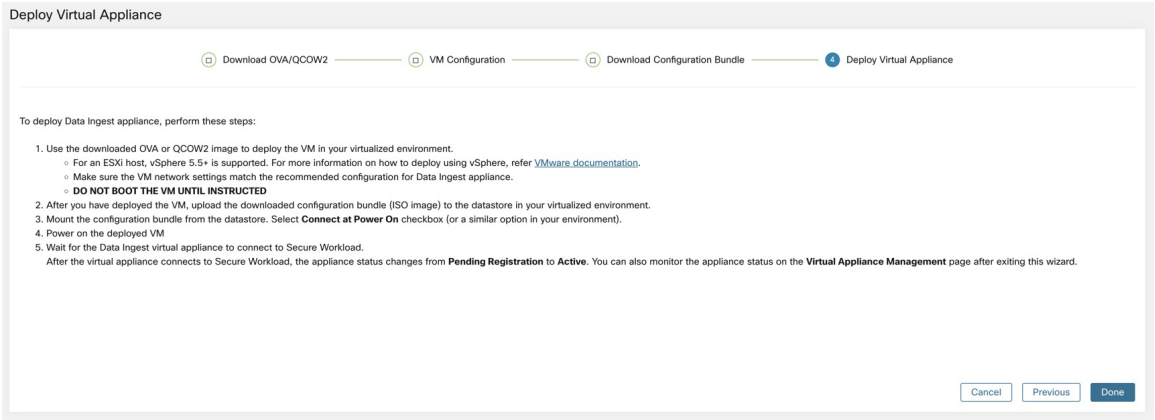
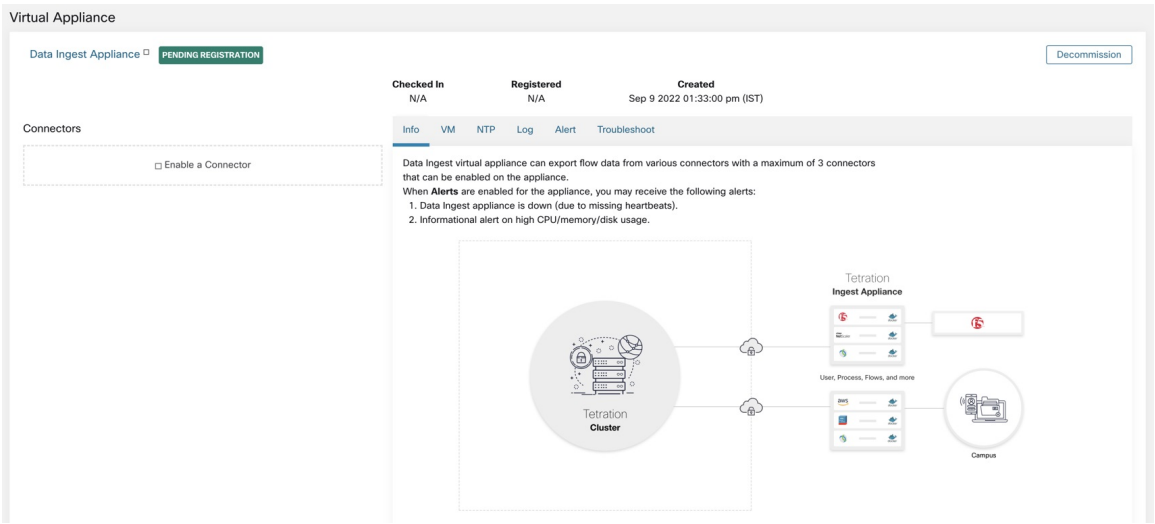


図 *54 :* 登録保留中の状態の *Secure Workload Ingest* アプライアンス



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）。

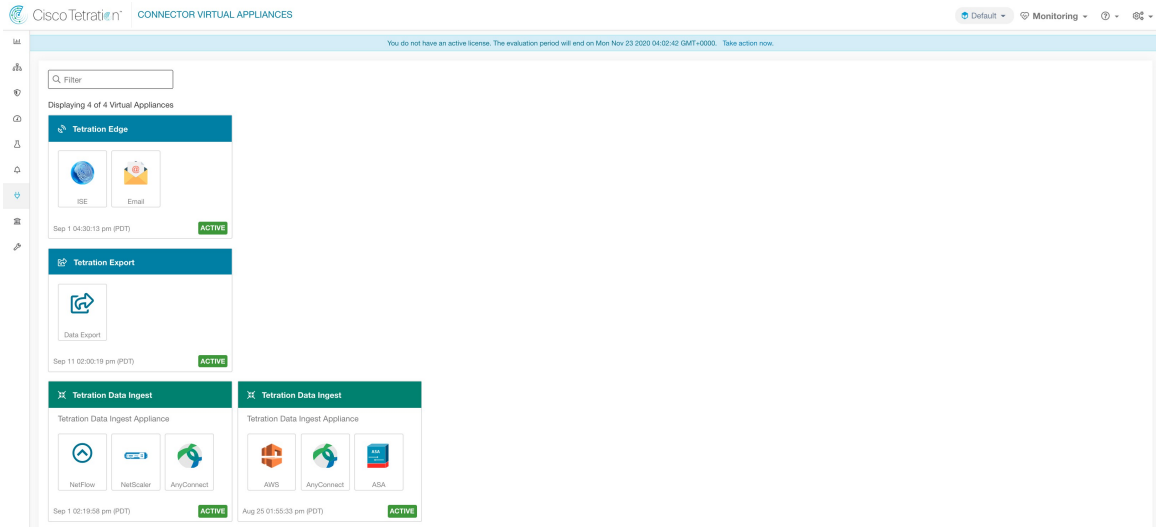
6. **アプライアンスの準備**：アプライアンス管理メッセージが送受信されるKafkaトピックの証明書バンドルをコピーします。
7. **アプライアンスコントローラのインストール**：*tet-controller*サービスとして*supervisord*によって管理されるアプライアンスコントローラをインストールして起動します。

*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 *55* : 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 *56 : 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 *57 : 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 *58 : 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 *59 : 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 *60 : 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 *61 :* 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。

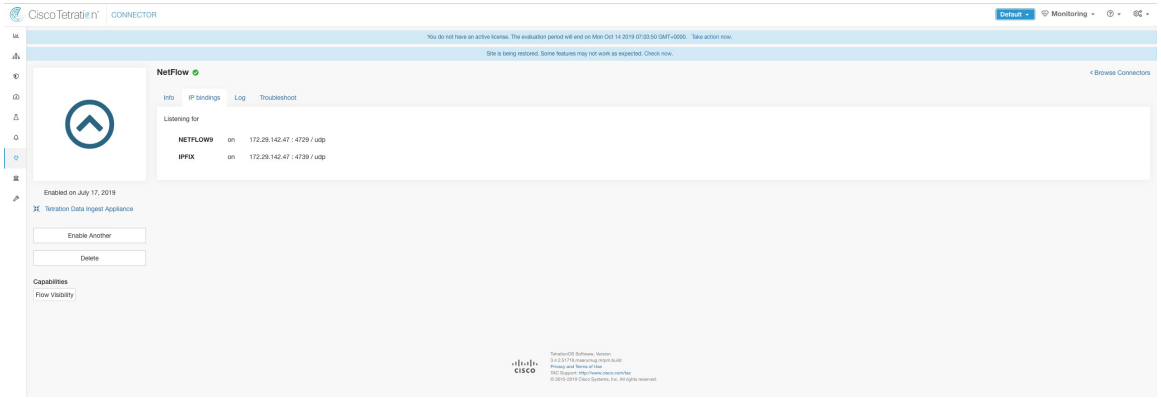
1. **登録**：コネクタを Cisco Secure Workload に登録します。コネクタが登録されて有効と表示されるまで、設定の更新をコネクタにプッシュすることはできません。Secure Workload は、コネクタの登録要求を受信すると、コネクタの状態を有効に更新します。
2. **コネクタの設定の更新**：コネクタの設定の更新をテストして適用します。詳細については、「コネクタおよび仮想アプライアンスの構成管理」を参照してください。
3. **コネクタのトラブルシューティングコマンド**：コネクタサービスの問題をトラブルシューティングおよびデバッグするために、コネクタサービスで許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。

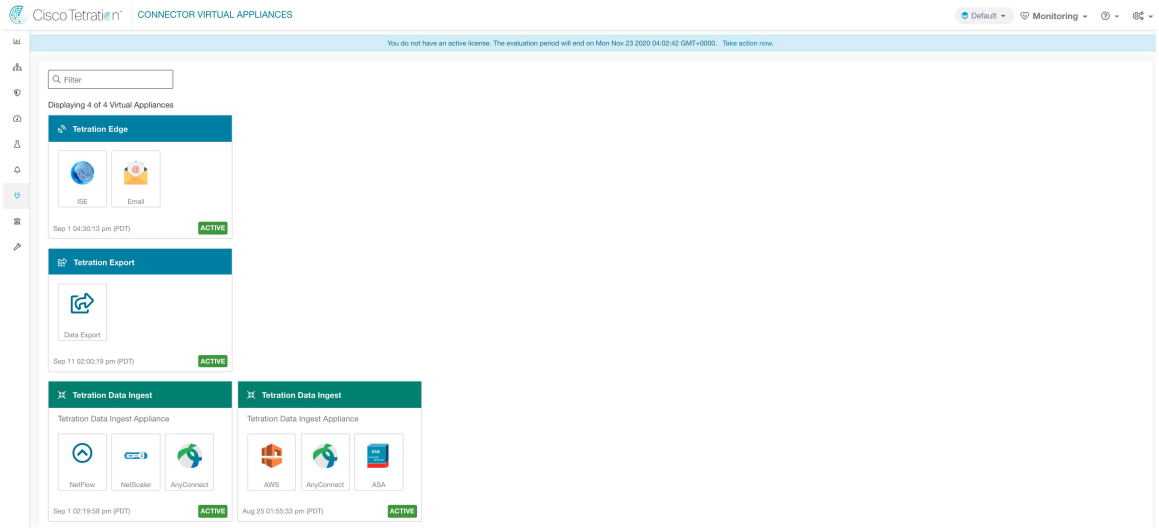
コネクタの詳細：コネクタをクリックすると、コネクタの詳細情報を取得できます。このページには、ポートバインディングが（存在する場合）表示されます。ポートバインディングは、テレメトリデータを正しい IP とポートに送信するようにアップストリーム ネットワーク要素を設定するために使用できます。

図 *62 :* コネクタの詳細



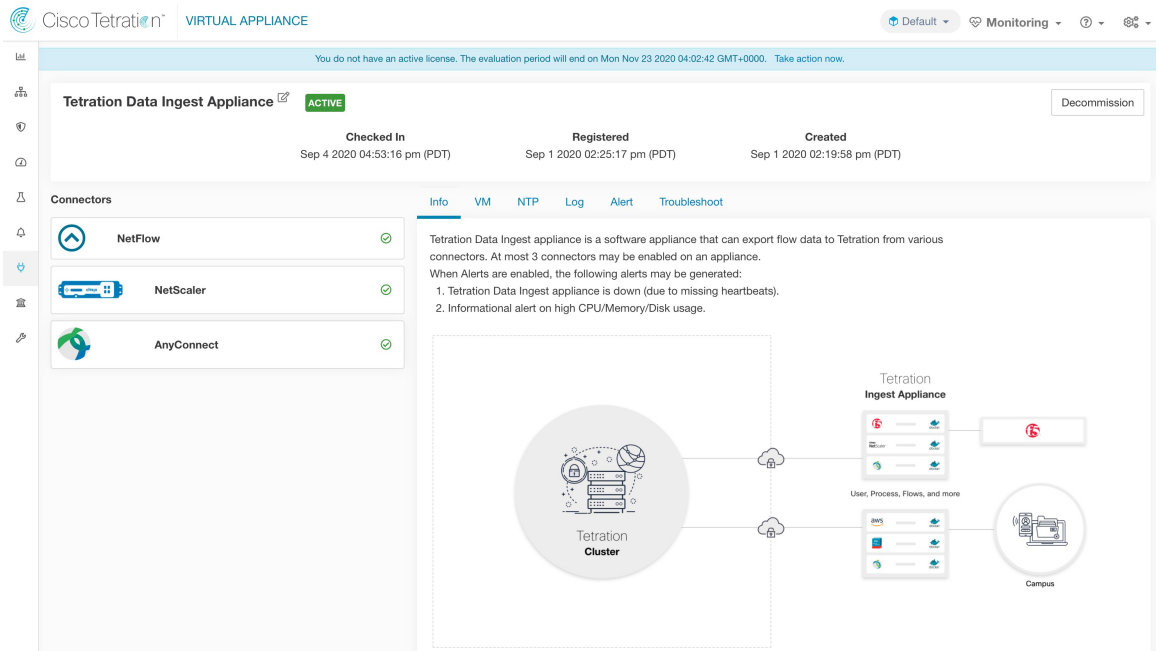
展開済みの仮想アプライアンス：展開済みの仮想アプライアンスのリストは、**[管理（Manage）]** > **[仮想アプライアンス（Virtual Appliances）]** にあります。

図 *63* : 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 *64* : 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 *65 : NTP* 設定のテスト中にエラーが発生

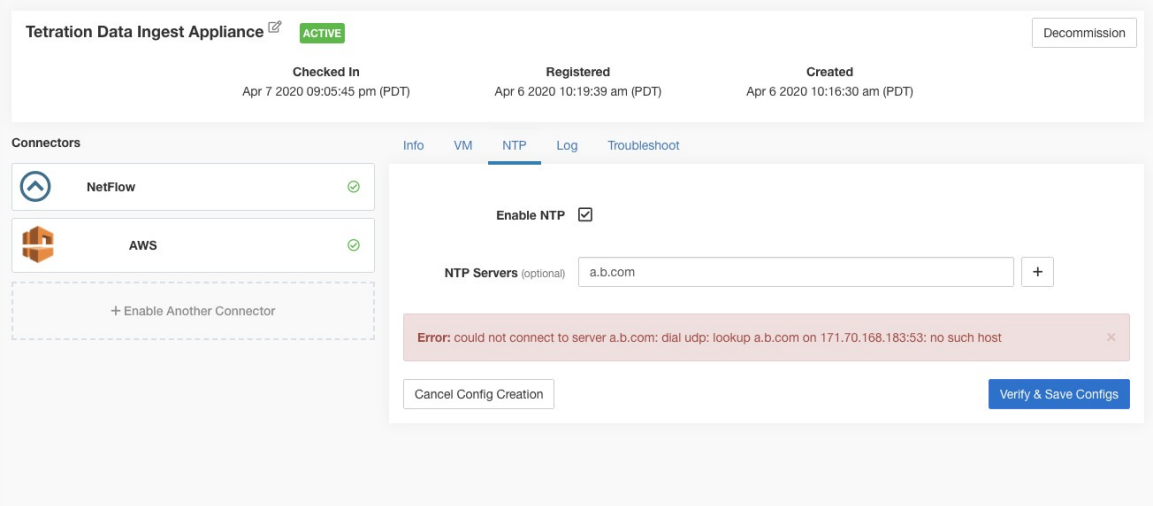


図 *66* : 有効な *NTP* サーバーを使用した *NTP* 設定

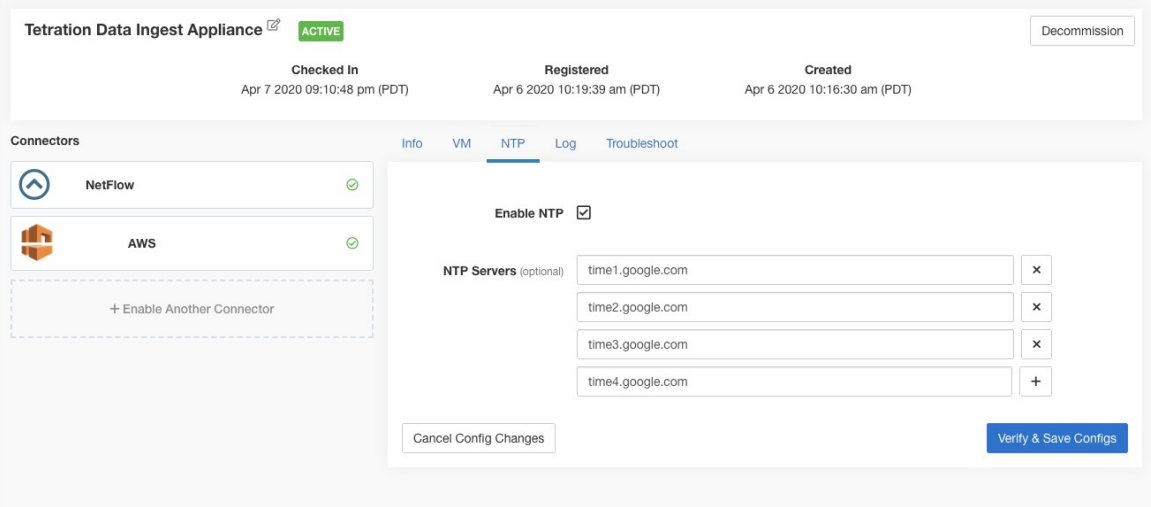
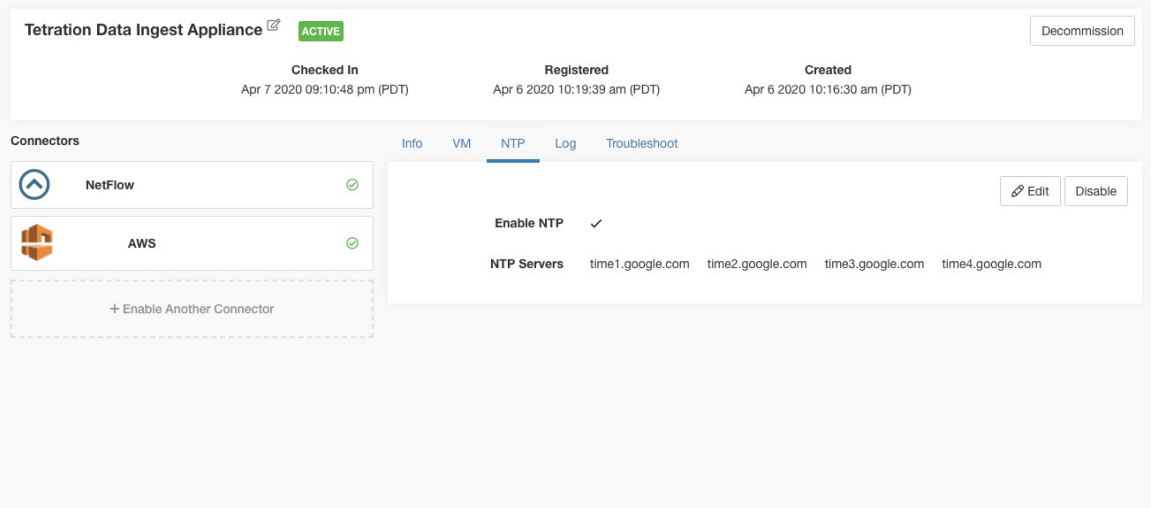


図 *67* : 検証および適用済みの *NTP* 設定



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

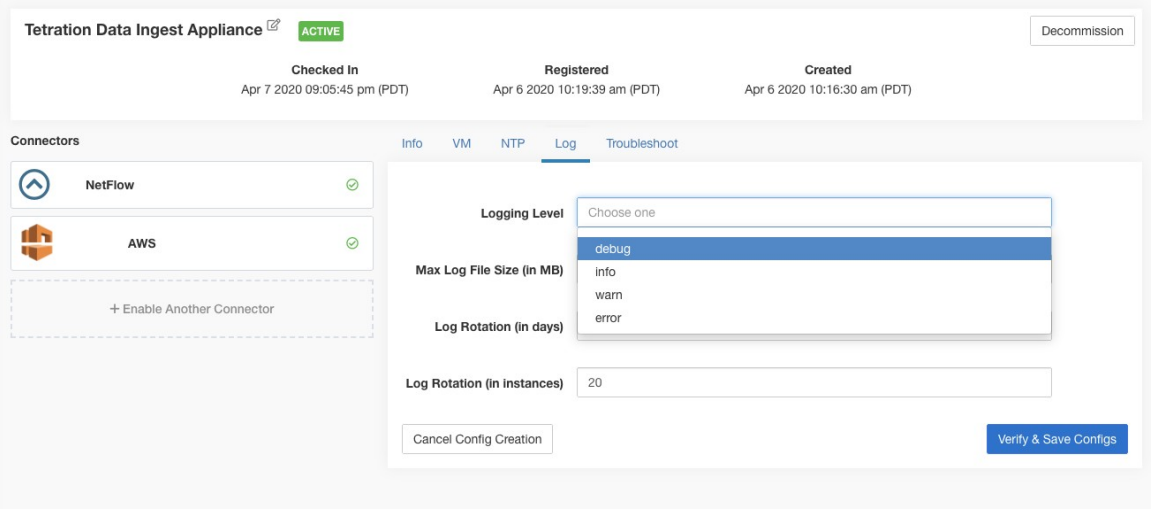
テスト：運用なし。

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 *68 :* 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。

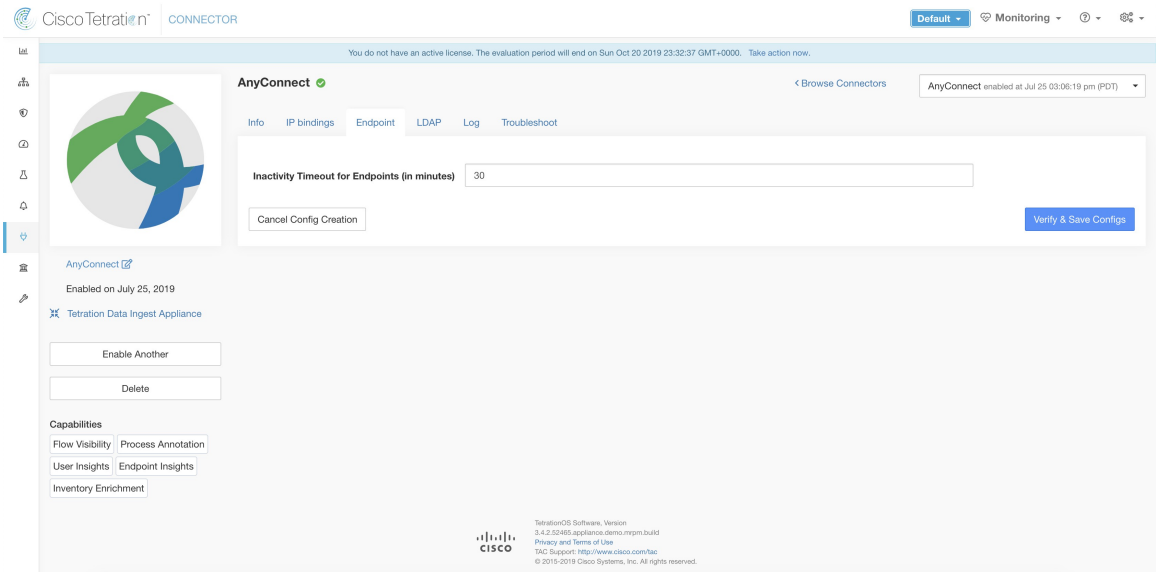
テスト：運用なし。

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 *69 : 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：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：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：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。

許可されている 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	string	AWS と通信するための AWS アクセスキー ID
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 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	string	Kinesis ストリームの名前
ストリームパーティション	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：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：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：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。

許可されている Secure Workload 仮想アプライアンス：なし

許可されたコネクタ：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 インスタンスの構成

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：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：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。

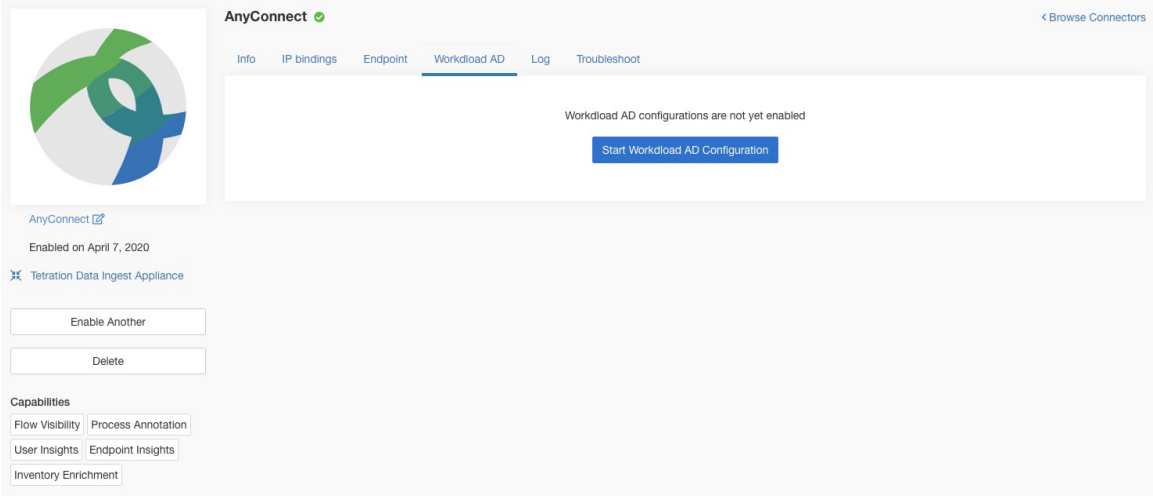
LDAPへの接続方法と基本DNの設定があれば、LDAP内のユーザーの属性を検出できます。これらの検出された属性は、UIでユーザーに表示できます。ユーザーは、検出されたこれらの属性から、ユーザー名に対応する属性と、ユーザー名ごとにLDAPから収集する最大6つの属性のリストを選択します。その結果、LDAP属性を手動で設定する必要がなくなり、エラーが減少します。

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。

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 *70 : LDAP* 設定の検出の開始

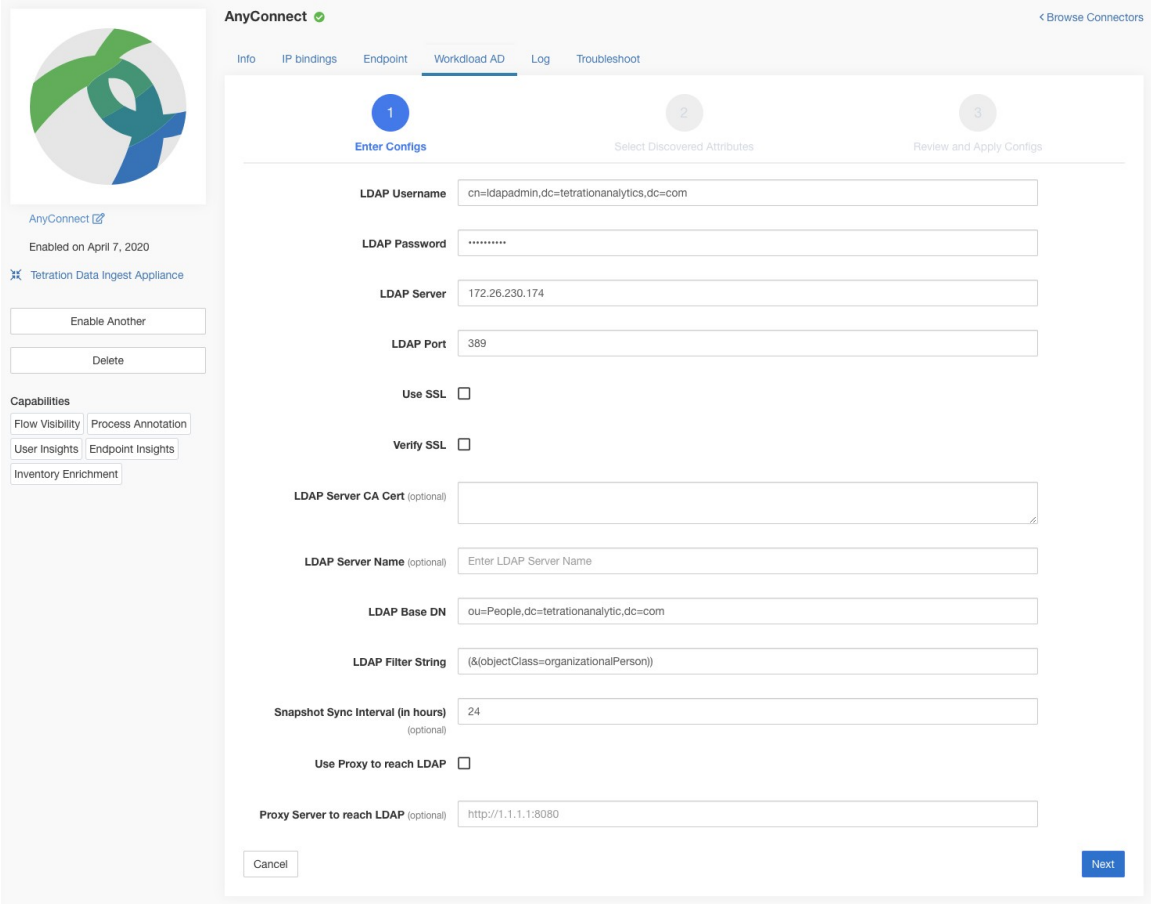


ステップ2 基本の LDAP 設定の指定

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。
[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

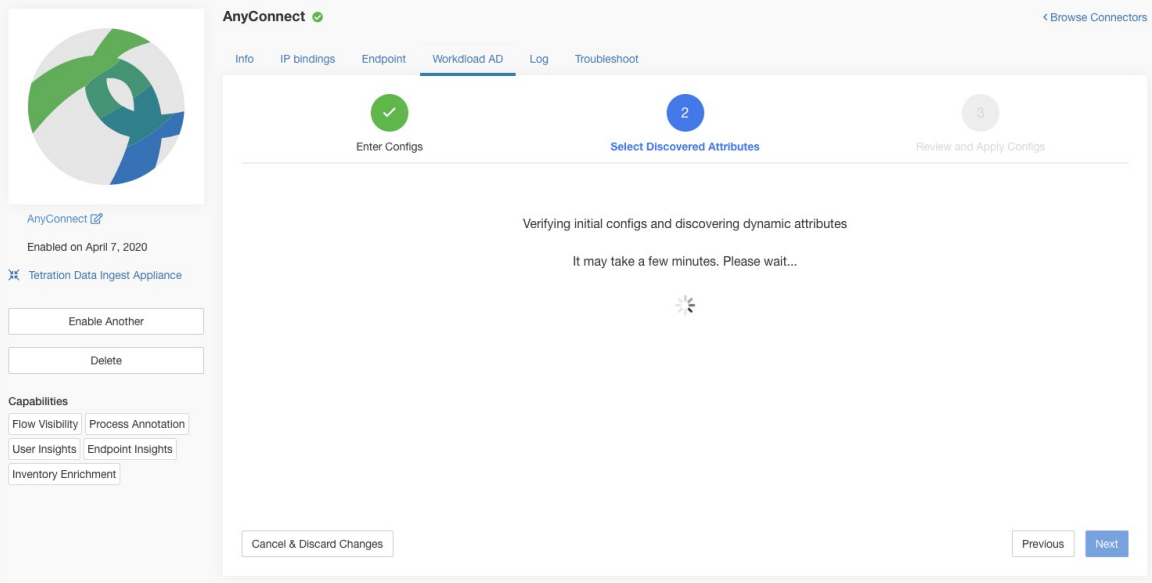
図 *71* : 初期 *LDAP* 設定



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 *72 : 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 *73* : 検出された *LDAP* 属性

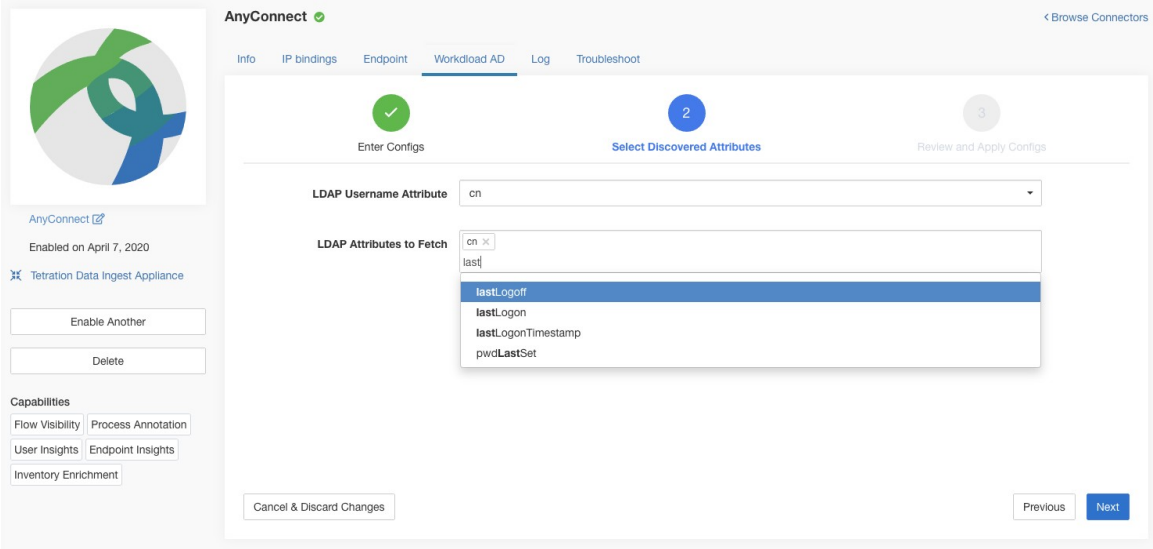
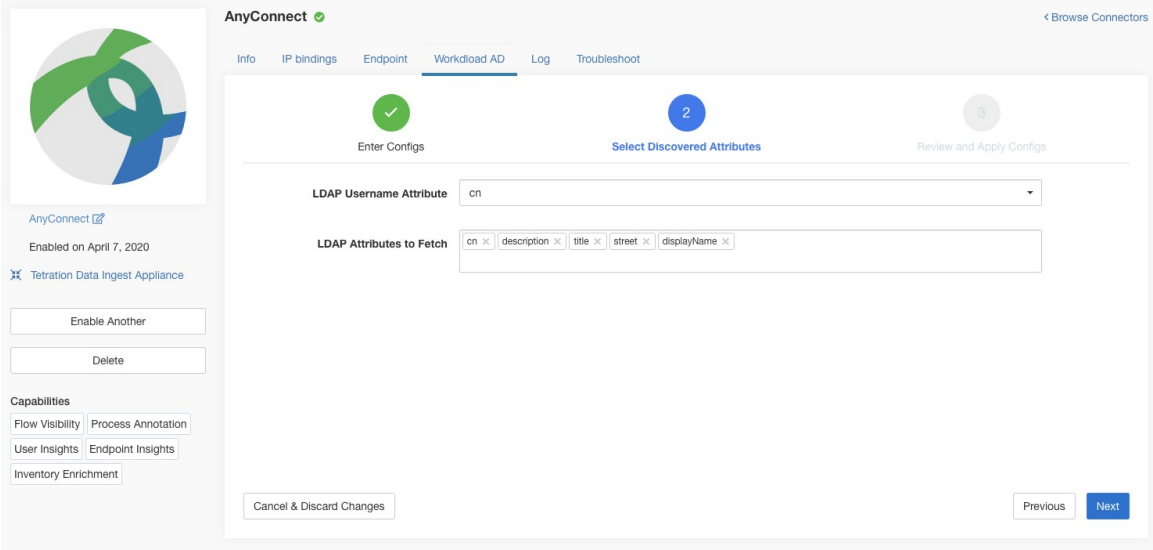


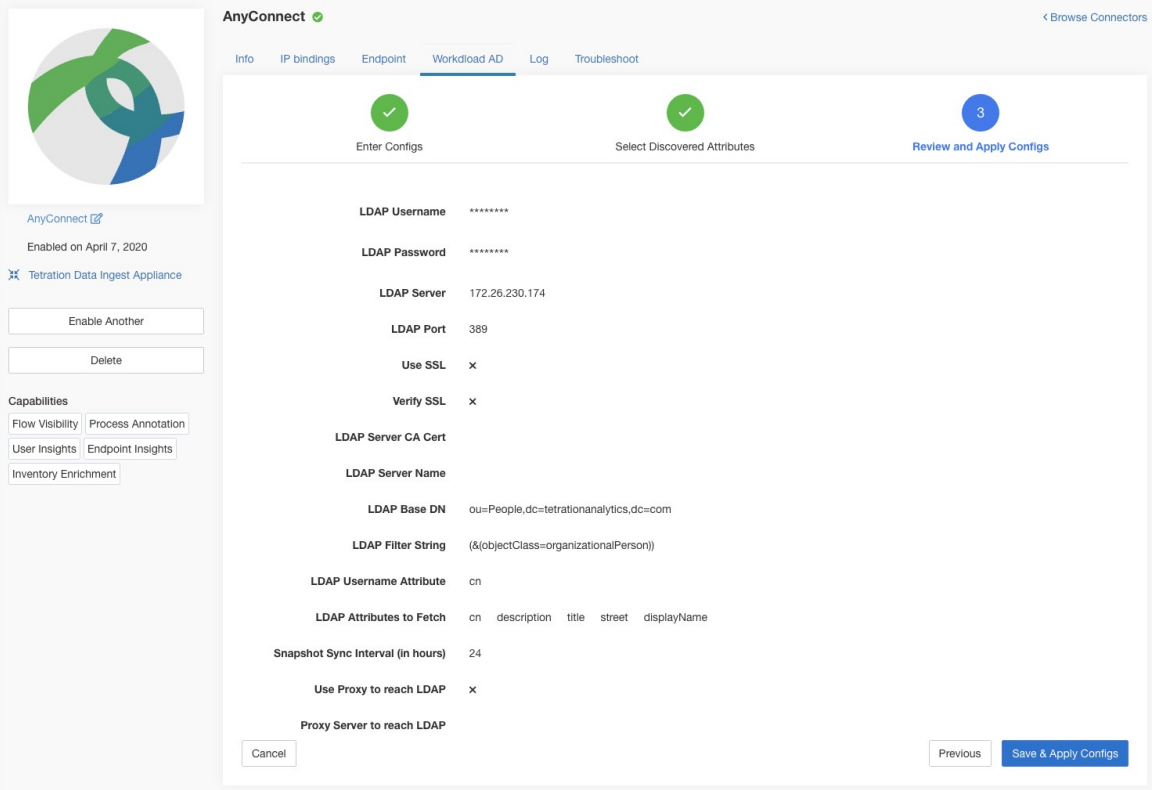
図 *74* : ユーザー名属性およびユーザー名ごとに収集する属性を特定



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 *75 : 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。

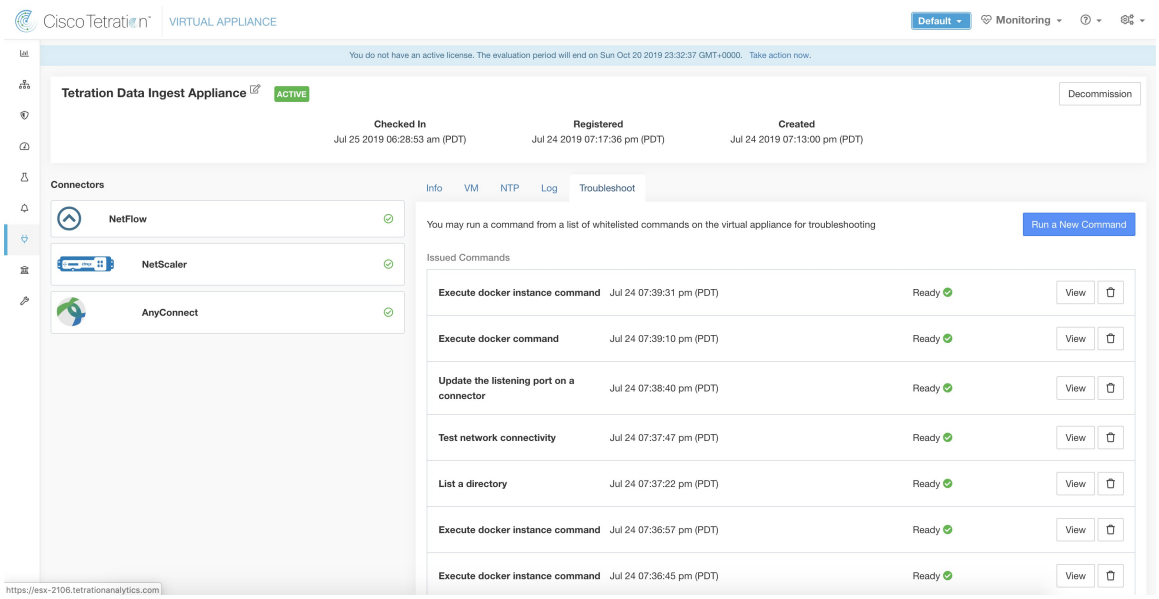
ERSPAN 仮想アプライアンス：トラブルシューティングの詳細については、ERSPAN アプライアンスのページを参照してください。

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 *76 : 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

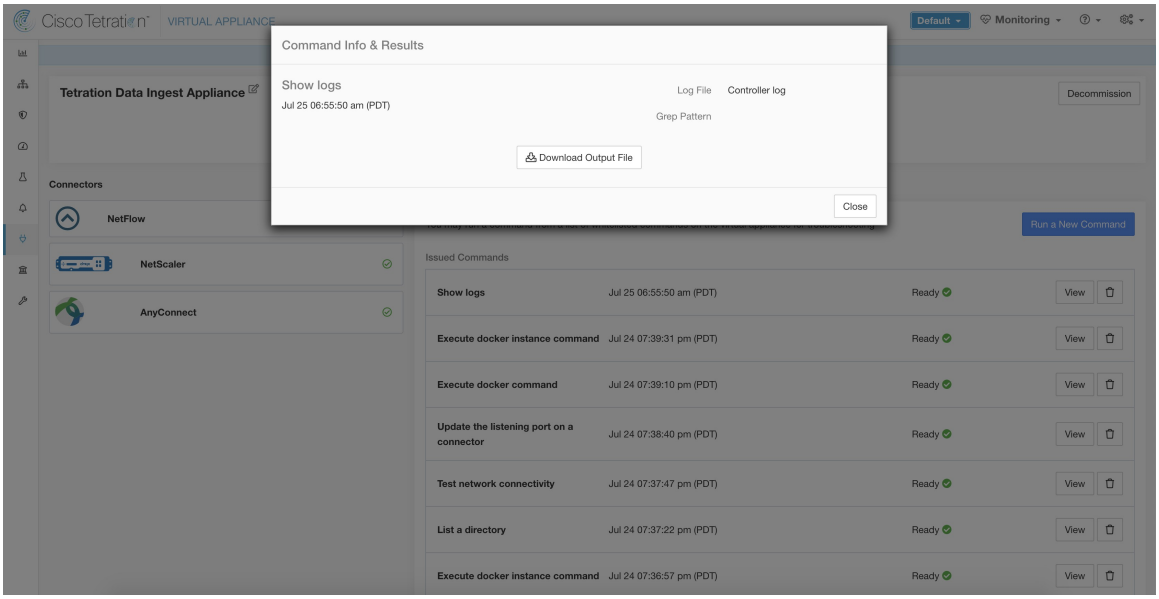
ンドを送信します。アプライアンスまたはコネクタサービスのコントローラから結果が返されます（最後の 5000 行の末尾）。結果が Cisco Secure Workload で利用可能になると、ファイルをダウンロードするためのダウンロードボタンが表示されます。

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、電子メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *77 : 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

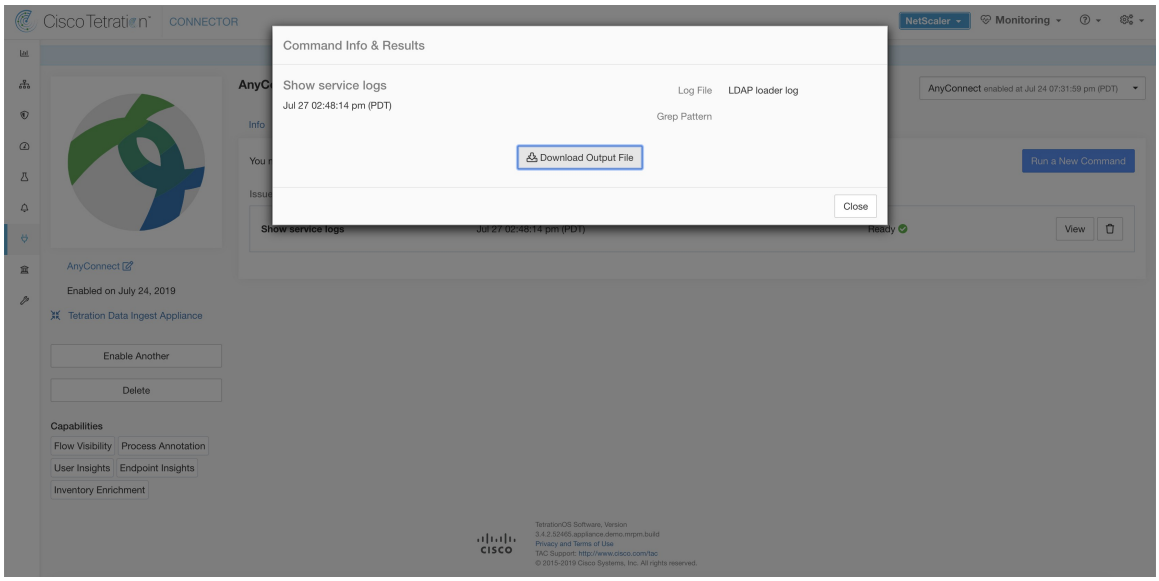
サービスログファイルの内容を表示し、オプションでファイルをgrep して指定されたパターンを見つけます。Secure Workload は、コマンドが発行されたアプライアンスまたはコネクタにコマンドを送信します。アプライアンスまたはコネクタサービスのコントローラから結果が返されます（最後の 5000 行の末尾）。結果が Cisco Secure Workload で利用可能になると、ファイルをダウンロードするためのダウンロードボタンが表示されます。

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、Email、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *78: 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、

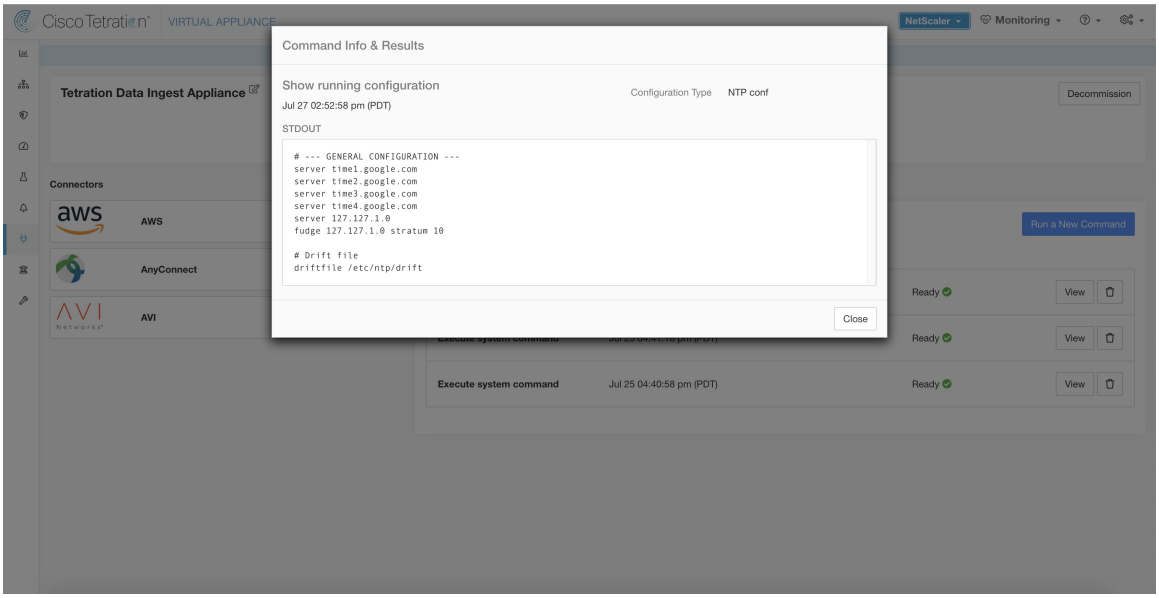
結果を返します。結果が Cisco Secure Workload で利用可能な場合、構成の内容がテキストボックスに表示されます。

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、Email、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *79 : 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

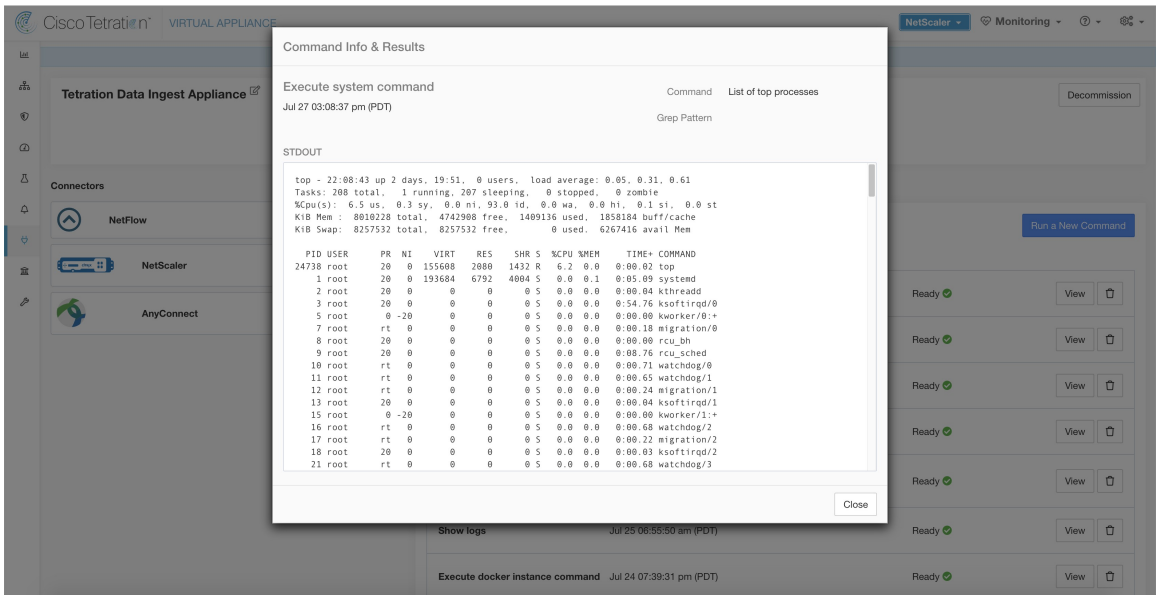
システムコマンドを実行し、指定したパターンの grep を実行します（オプション）。アプライアンスまたはコネクタサービスのコントローラから結果が返されます（最後の 5000 行の末尾）。必要に応じて、grep パターンを引数として指定でき、そのパターンに応じて出力がフィルタリングされます。結果が Cisco Secure Workload で利用可能な場合、テキストボックスに表示されます。

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、Email、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *80 : 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

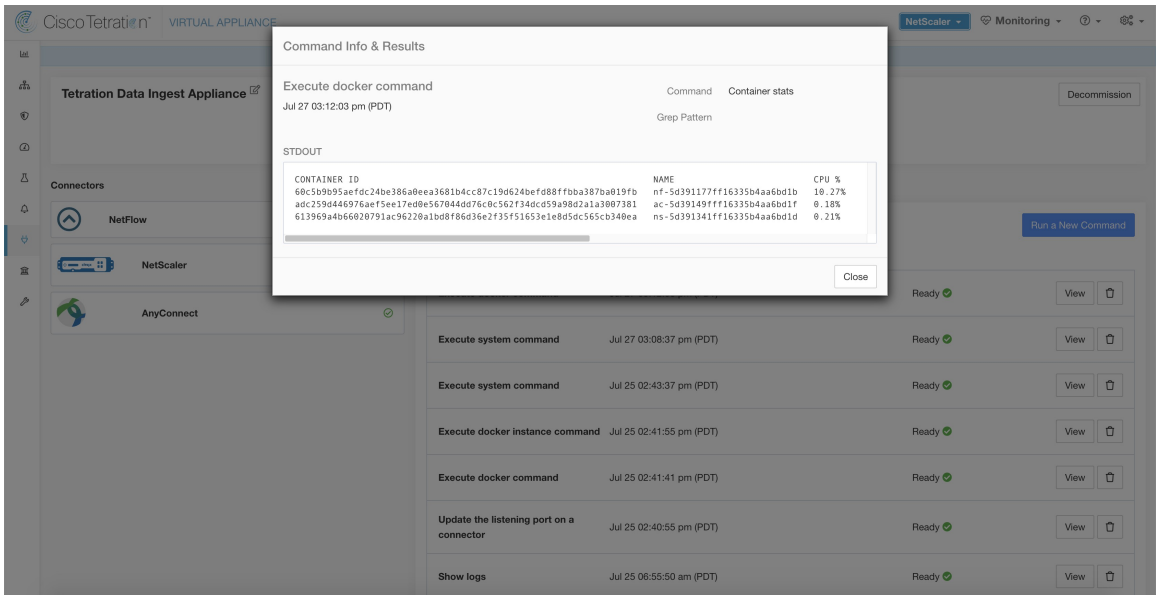
Docker コマンドを実行し、指定したパターンの grep を実行します（オプション）。コマンドは、アプライアンスコントローラによってアプライアンス上で実行されます。結果は、最後の 5000 行まで表示されます。必要に応じて、grep パターンを引数として指定でき、そのパターンに応じて出力がフィルタリングされます。結果が Cisco Secure Workload で利用可能な場合、結果はテキストボックスに表示されます。

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 *81 : 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

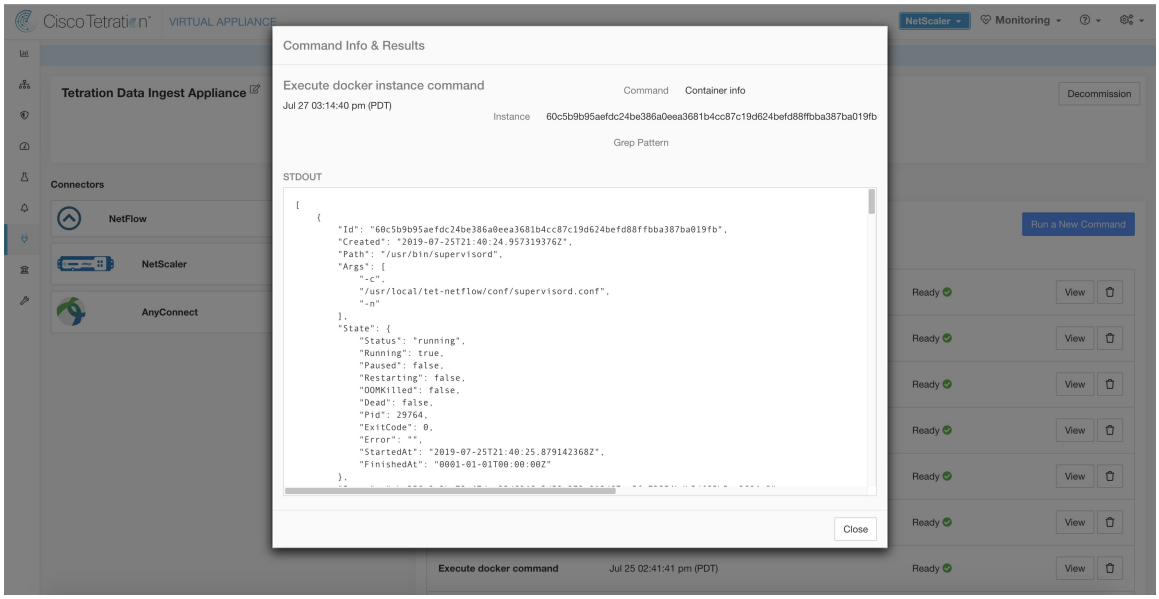
Docker リソースの特定のインスタンスで Docker コマンドを実行します。インスタンス ID は、Docker コマンドの表示を使用して取得できます。コマンドは、アプライアンスコントローラによってアプライアンス上で実行されます。結果は、最後の 5000 行まで表示されます。必要に応じて、grep パターンを引数として指定でき、そのパターンに応じて出力がフィルタリングされます。結果が Cisco Secure Workload で利用可能な場合、テキストボックスに表示されます。

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	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 *82 : 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

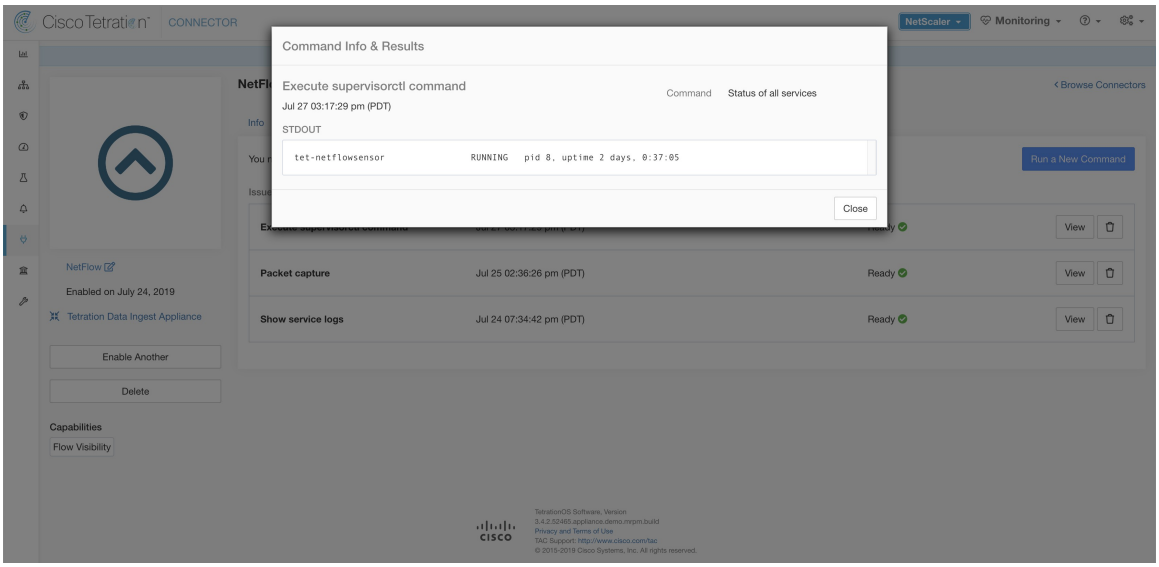
supervisorctl コマンドを実行し、結果を返します。Secure Workload はコマンドが発行されたアプライアンスまたはコネクタにコマンドを送信します。アプライアンスまたはコネクタサービスのコントローラから結果が返されます。結果は、Cisco Secure Workload で利用可能な場合はテキストボックスに表示されます。

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、Email、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

図 *83* : 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

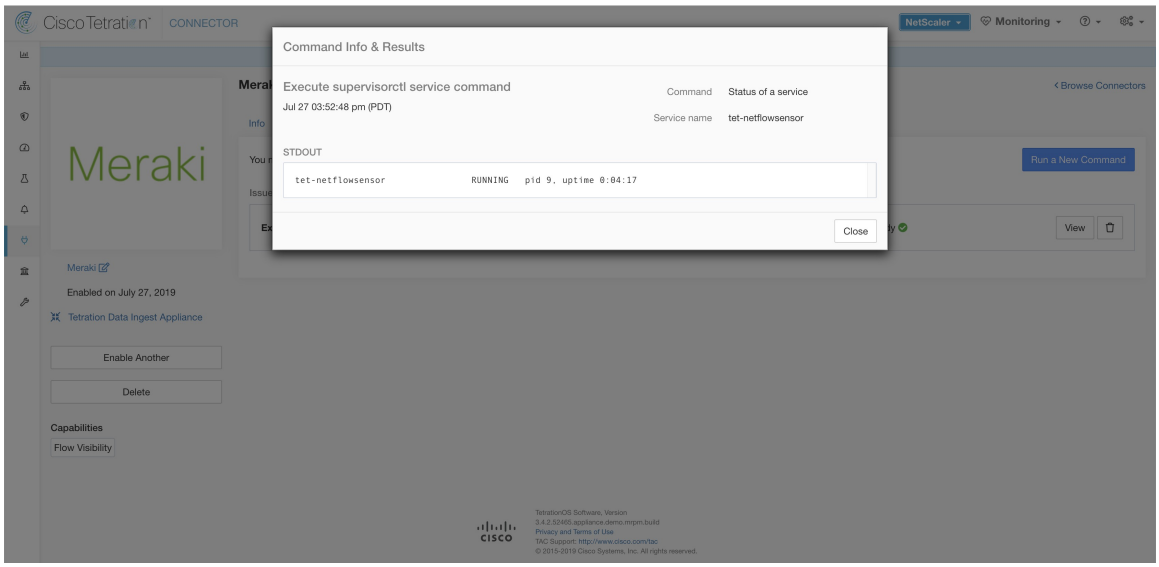


スーパーバイザサービスのコマンドの表示

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 *84 : 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、電子メール、Slack、PagerDuty、Kinesis、ISE、ASA、Meraki。

ネットワーク接続コマンド

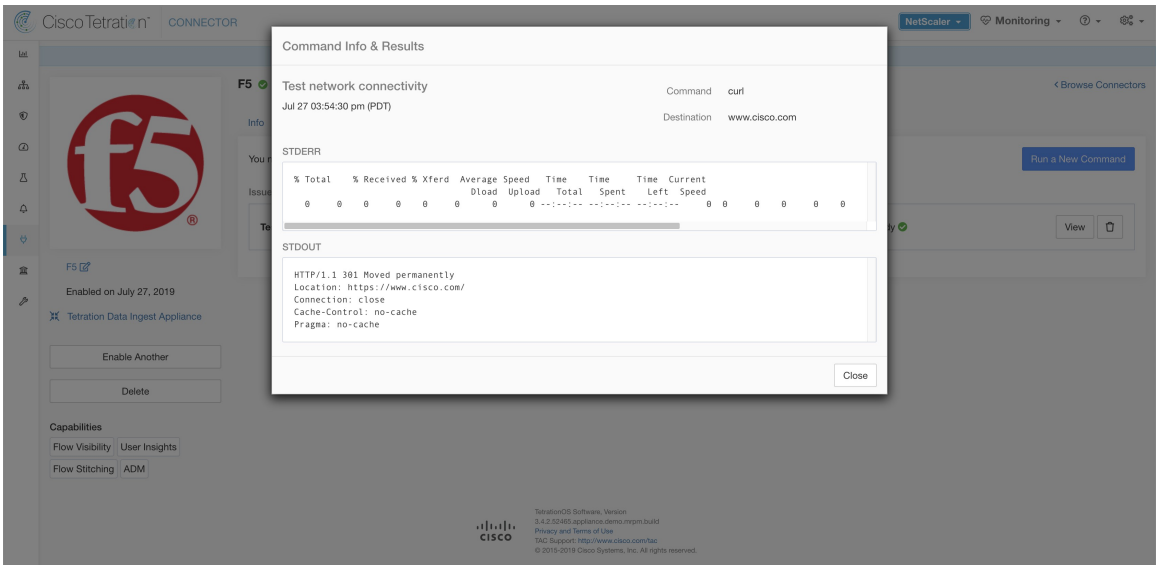
アプライアンスやコネクタからネットワーク接続をテストします。コマンドは、アプライアンスコントローラによってアプライアンス上で実行されます。結果が Cisco Secure Workload で利用可能になると、テキストボックスに表示されます。

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 *85 : 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

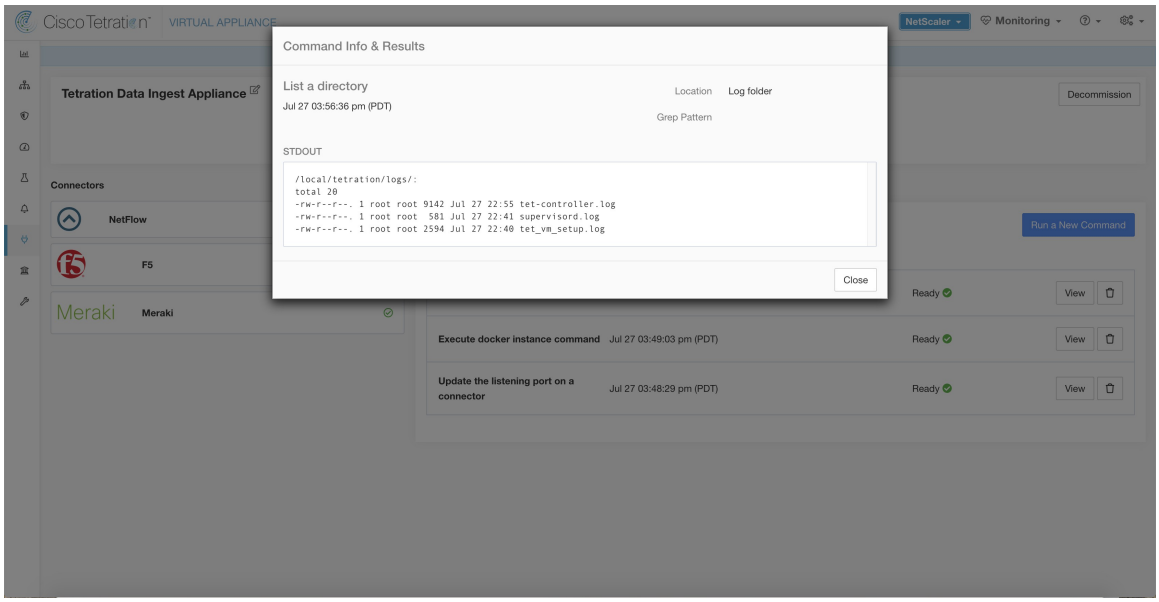
アプライアンスの既知の場所にあるファイルを一覧表示します。（オプション）指定されたパターンの grep。Secure Workload は、コマンドが発行されたアプライアンスにコマンドを送信します。アプライアンスのコントローラが結果を返します。結果が Cisco Secure Workload で利用可能な場合、結果はテキストボックスに表示されます。

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	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。
パターンの Grep	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 *86 : 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

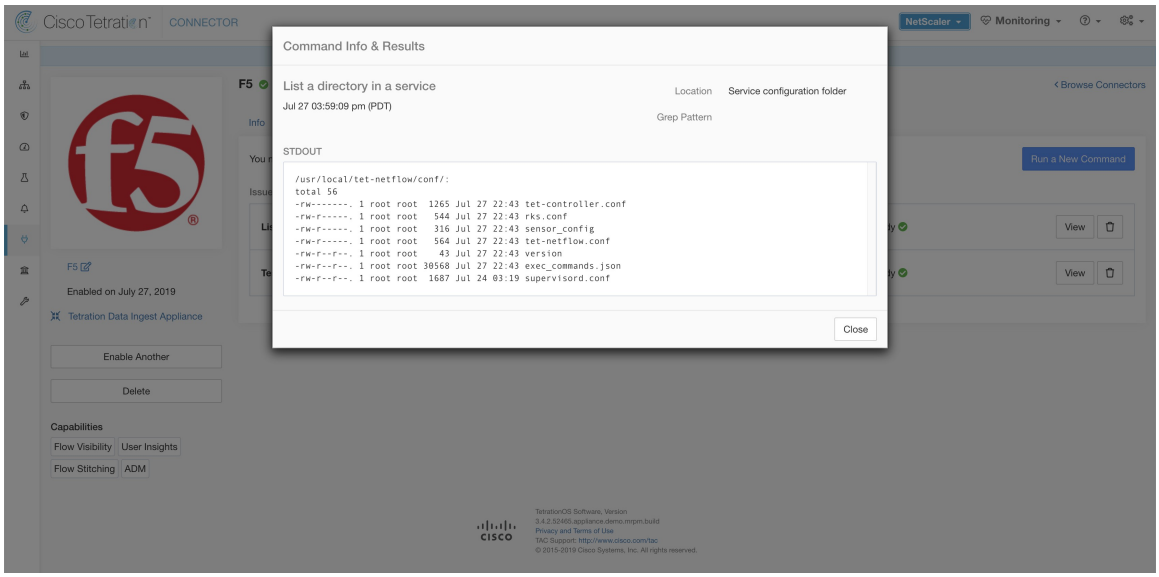
コネクタサービスの既知の場所にあるファイルを一覧表示します。（オプション）指定されたパターンの grep。Secure Workload は、コマンドが発行されたコネクタにコマンドを送信します。コネクタサービスのコントローラが結果を返します。結果が Cisco Secure Workload で利用可能な場合、結果はテキストボックスに表示されます。

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 *87 : 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

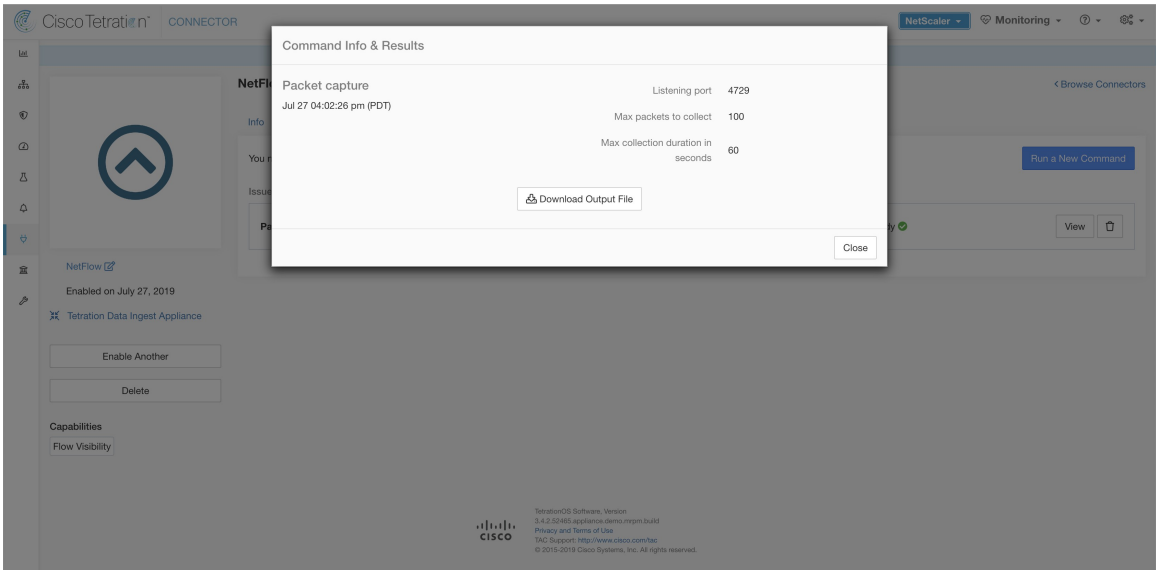
アプライアンスおよびコネクタで着信パケットをキャプチャします。Secure Workload はコマンドが発行されたアプライアンスおよびコネクタにコマンドを送信します。アプライアンスおよびコネクタサービスのコントローラは、パケットをキャプチャしてエンコードし、その結果を Cisco Secure Workload に返します。結果が Cisco Secure Workload で利用可能になると、ダウンロードボタンが表示され、ファイルを .pcap 形式でダウンロードできるようになります。

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 *88 : 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 *89 : 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

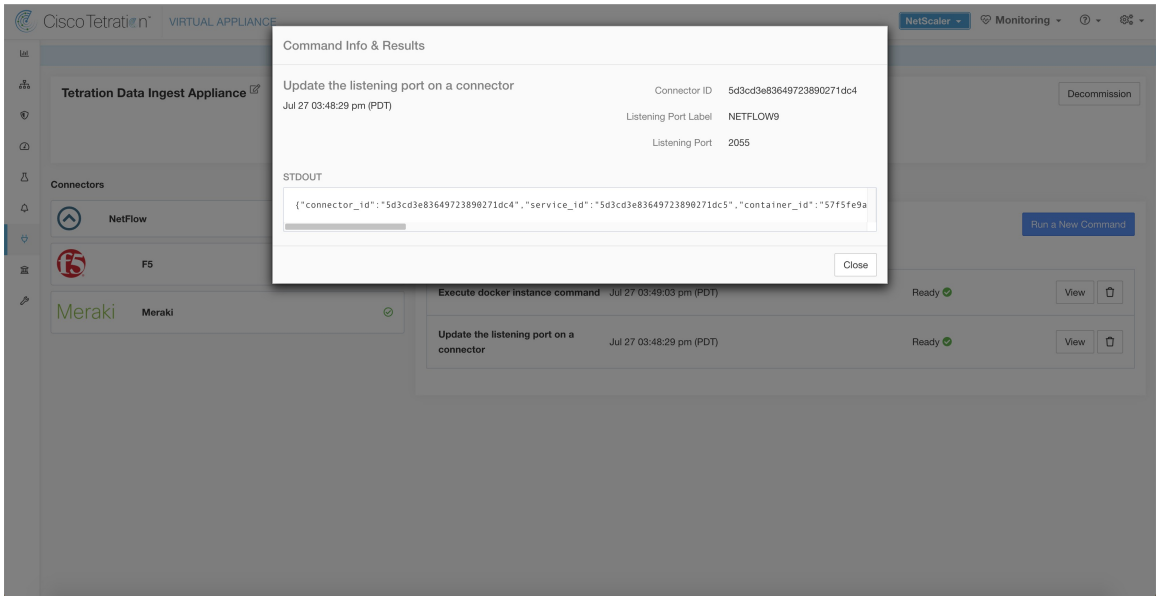
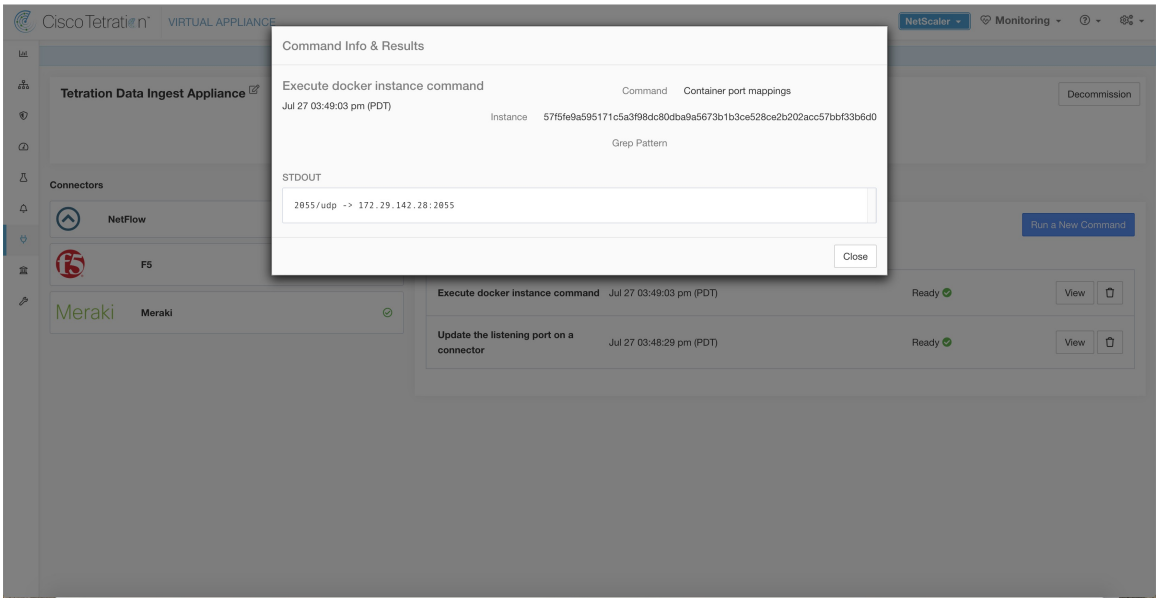


図 *90 : Secure Workload Ingest* アプライアンスで *Meraki* コネクタのポートマッピングを取得します



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。

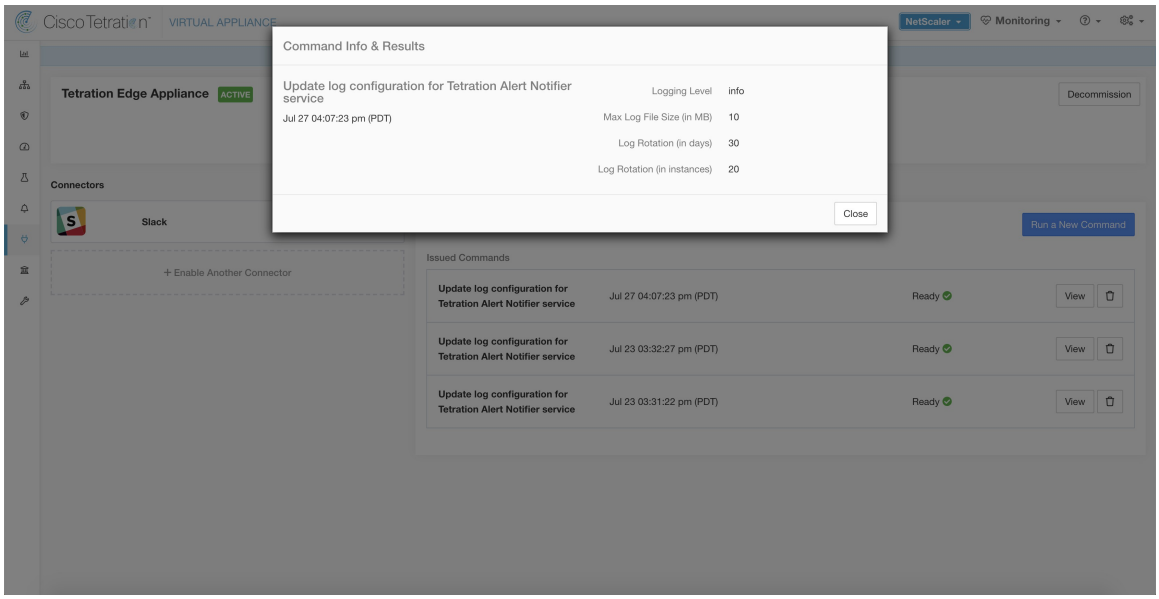
引数名	タイプ	説明
Logging level	dropdown	サービスで使用されるログレベル
	• *debug*	デバッグログレベル
	• *info*	情報ログレベル
	• *warn*	警告ログレベル
	• *error*	エラーログレベル
[最大ログファイルサイズ（MB単位）（Max log file size (in MB）]	number	ログ ローテーションが開始される前のログ ファイルの最大サイズ

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 *91 : 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

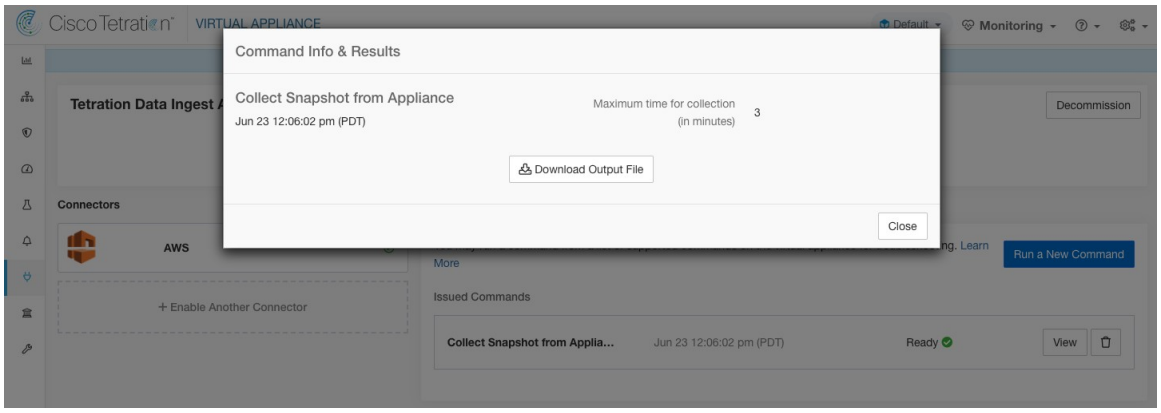
スナップショットに含まれるコマンド出力：

- 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 *92 : 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}

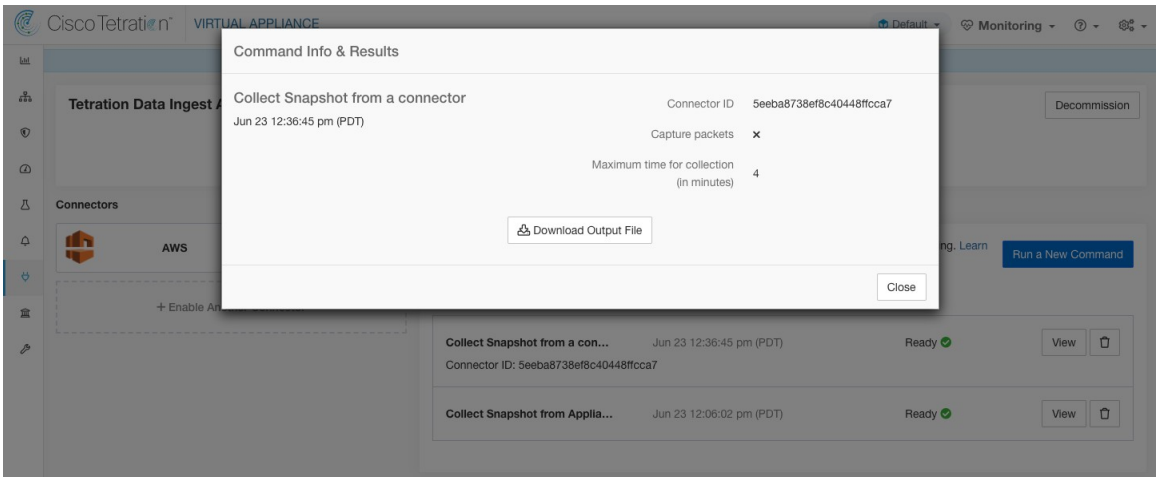
スナップショットに含まれるコマンド出力：

- 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 *93* : 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。

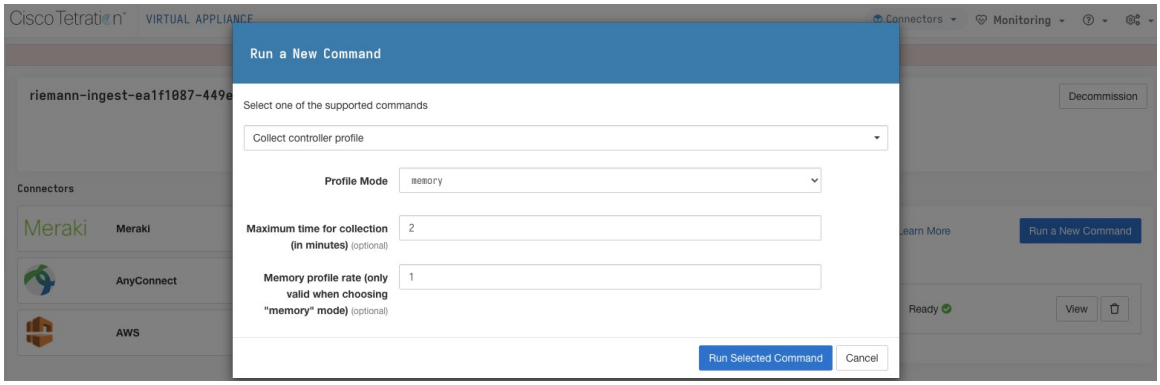
引数名	タイプ	説明
プロファイルモード	dropdown	プロファイリングモード。
	• *memory*	メモリ プロファイリング モード。
	• *cpu*	CPU プロファイリングモード。
	• *block*	ブロック プロファイリング モード。
	• *mutex*	Mutex プロファイリングモード。
	• *goroutine*	Goroutine プロファイリング モード。

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 *94 : 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

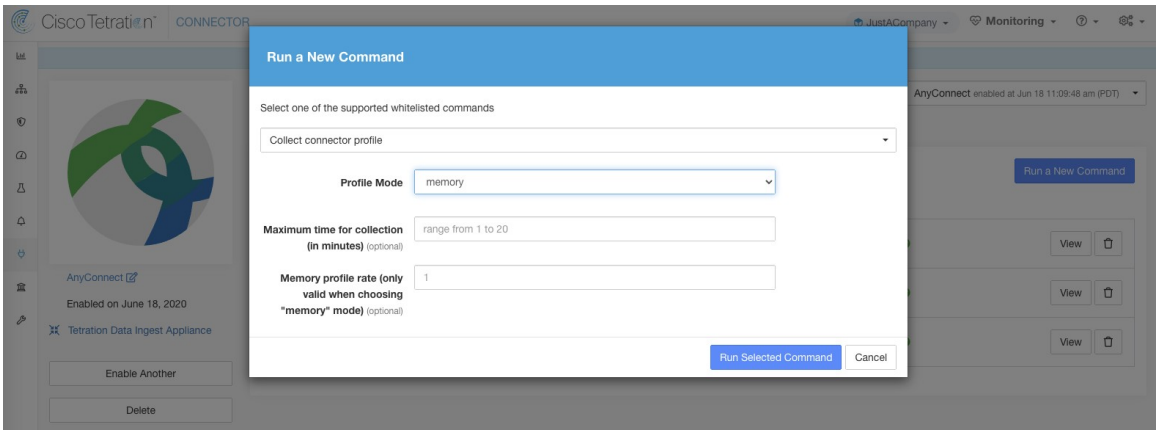
コネクタでのコネクタ プロセス プロファイリング結果を収集します。Secure Workload は、コマンドが発行されたコネクタにコマンドを送信します。サービスコントローラは、指定されたプロファイリングモードでコネクタサービスを再起動します。プロファイリング結果を収集した後、サービスコントローラは、通常モードでサービスを再起動し、結果を Cisco Secure Workload に送信します。結果が Cisco Secure Workload で利用可能になると、ダウンロードボタンが表示され、ファイルを `.tar.gz` 形式でダウンロードできるようになります。

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 *95 : 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

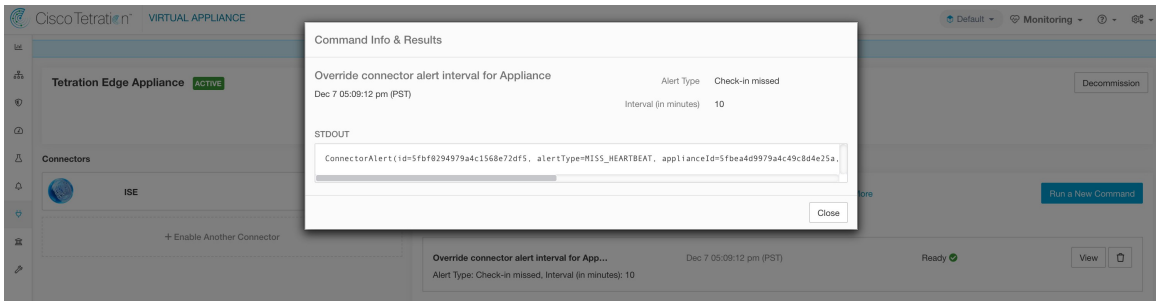
アプライアンスのデフォルトのコネクタアラート間隔をオーバーライドします。デフォルトで、Secure Workload では同じコネクタアラートが 1 日に 1 回だけ送信されるように制限されています。このコマンドは、管理者が 1 日 1 回の間隔では長すぎると考える場合に間隔をオーバーライドするためのものです。結果が Cisco Secure Workload で利用可能な場合、結果はテキストボックスに表示されます。

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	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 *96 : 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

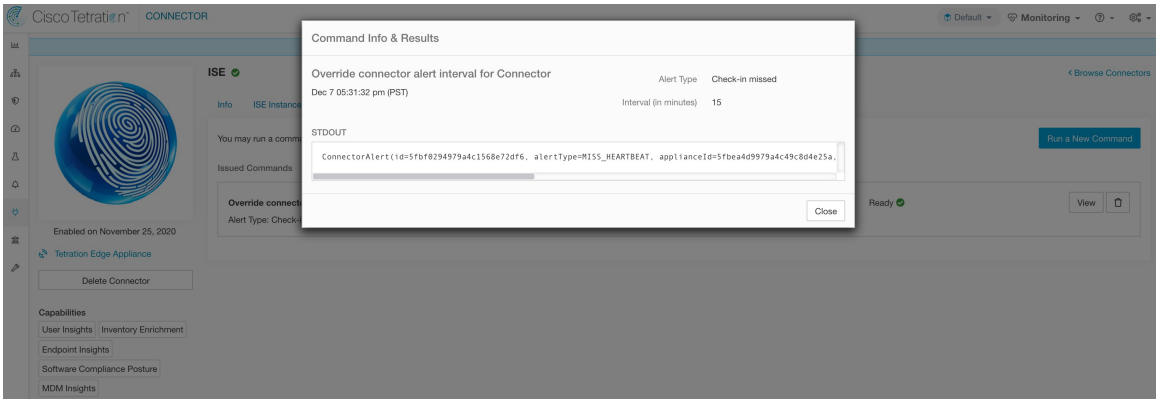
コネクタのデフォルトのコネクタアラート間隔をオーバーライドします。デフォルトで、Secure Workload では同じコネクタアラートが 1 日に 1 回だけ送信されるように制限されています。このコマンドは、管理者が 1 日 1 回の間隔では長すぎると考える場合に間隔をオーバーライドするためのものです。結果が Cisco Secure Workload で利用可能な場合、結果はテキストボックスに表示されます。

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 *97 : 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



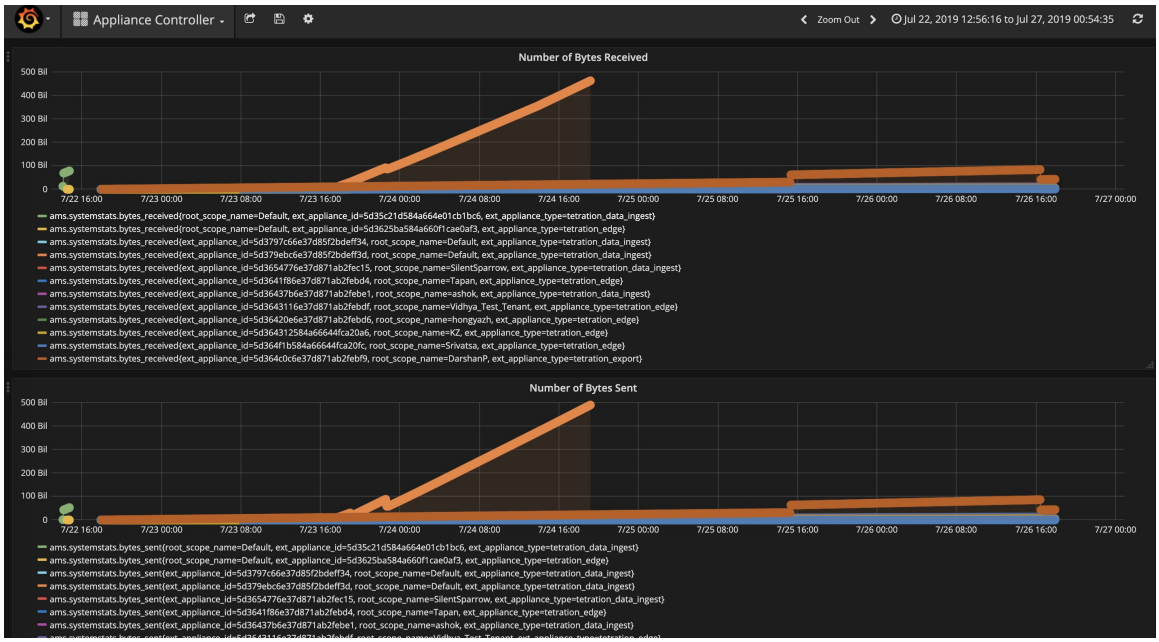
Hawkeye ダッシュボード

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。

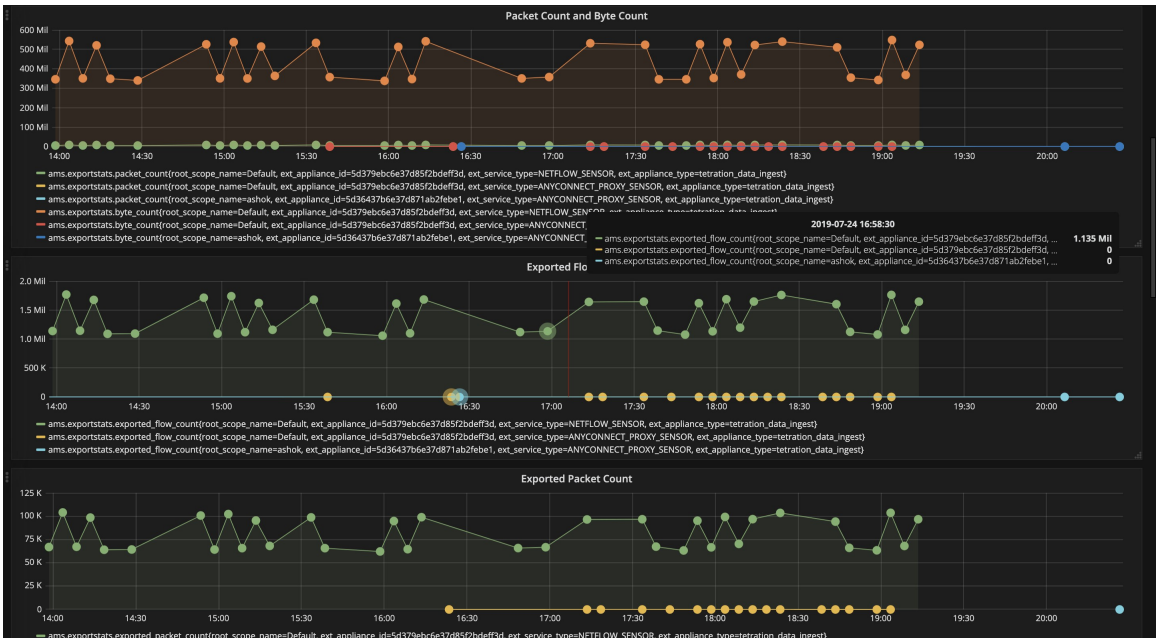
図 *98 :* アプライアンス コントローラ ダッシュボード



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。

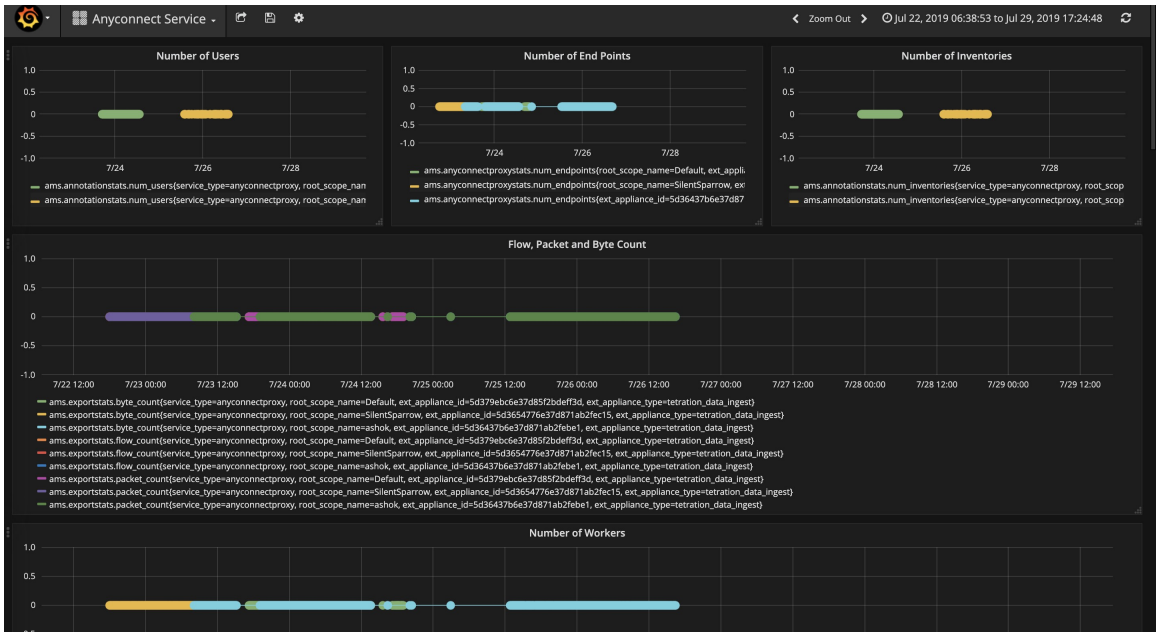
図 *99* : サービスダッシュボード



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。

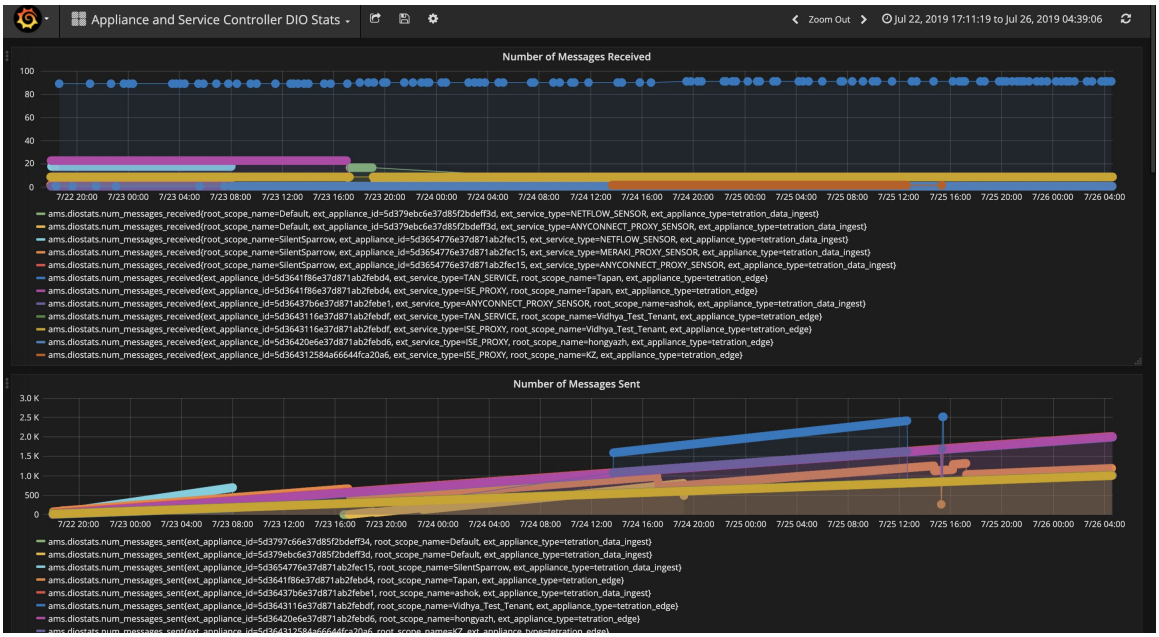
図 *100 : AnyConnect* ダッシュボード



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 *101* : 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 *102 : 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 *103 : 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 *104 :* 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 *105 : 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。

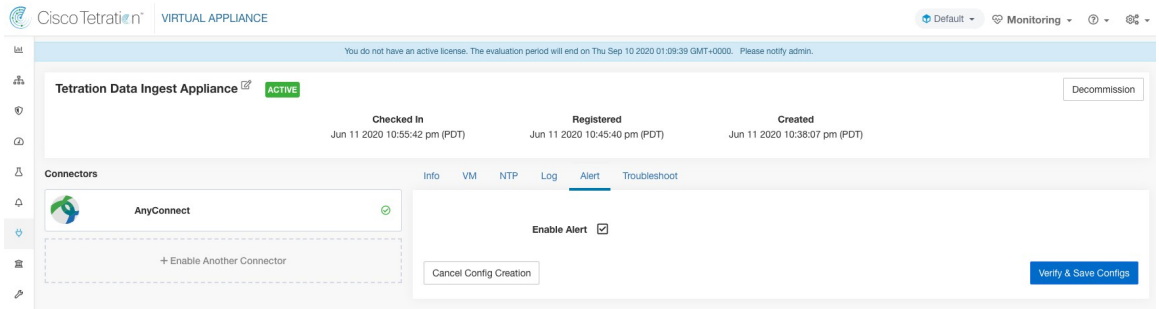
アラート設定

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。

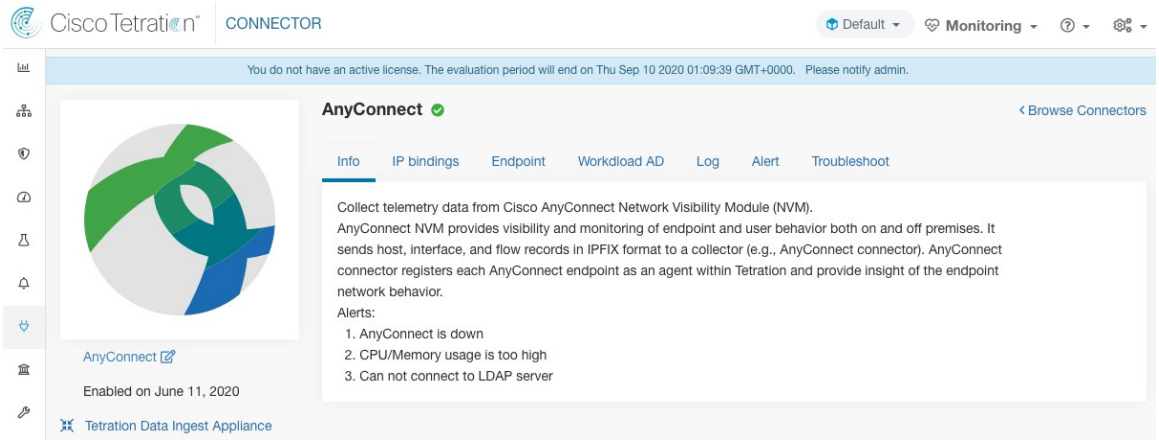
図 *106 : Secure Workload Data Ingest* アプライアンスでのアラート設定の表示



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 *107* : アラートリスト情報



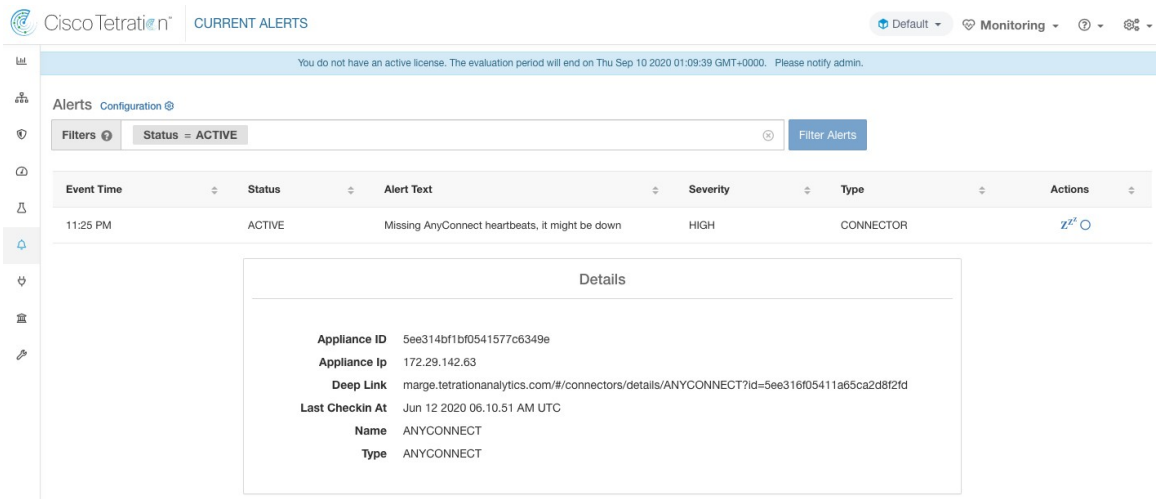
アプライアンス/コネクタダウン

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 *108* : 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

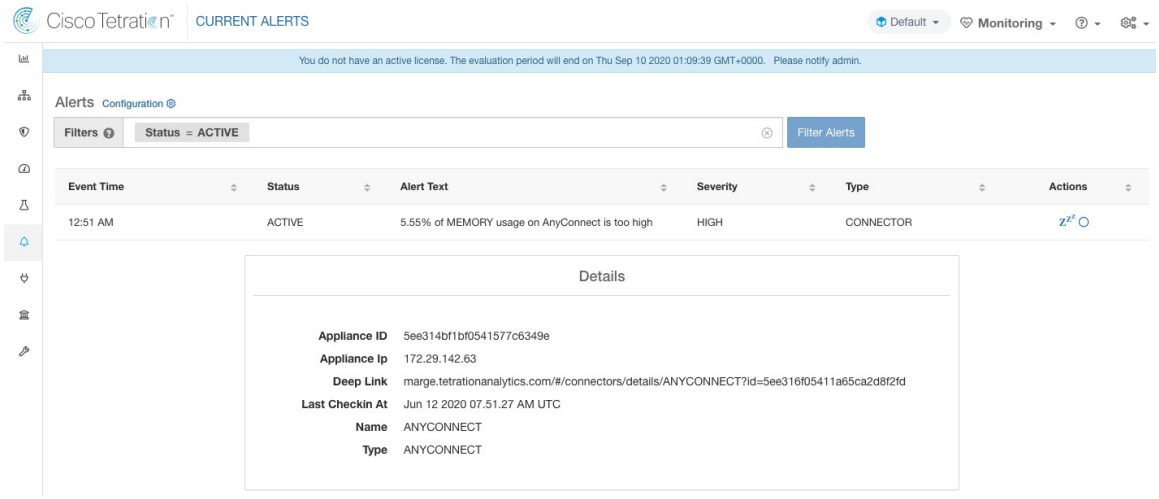
アプライアンスまたはコネクタシステムの使用率

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 *109*：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。

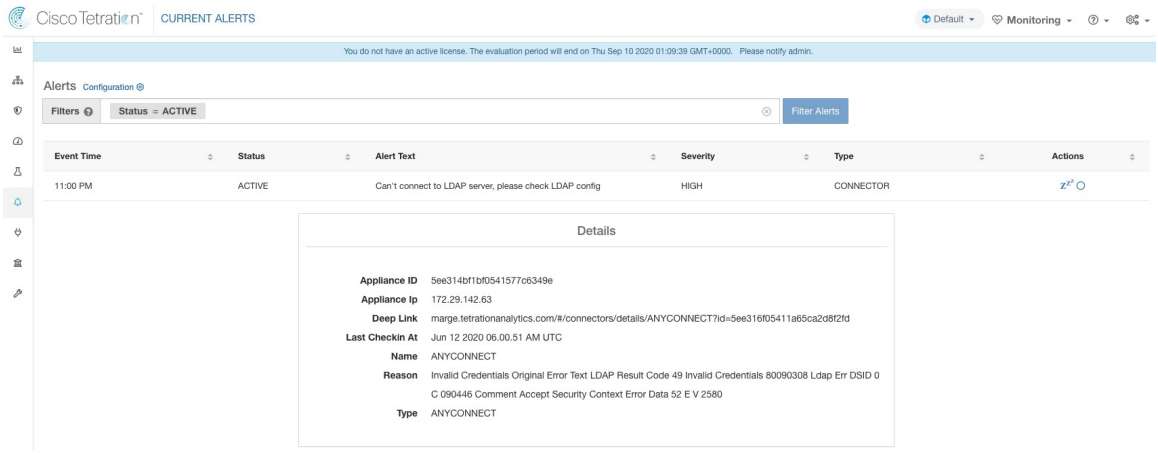
アラートテキスト：<Appliance/Connector> サーバーに接続できません。<Appliance/Connector> 構成を確認してください。

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 *110* : 構成ステータスエラーのアラート

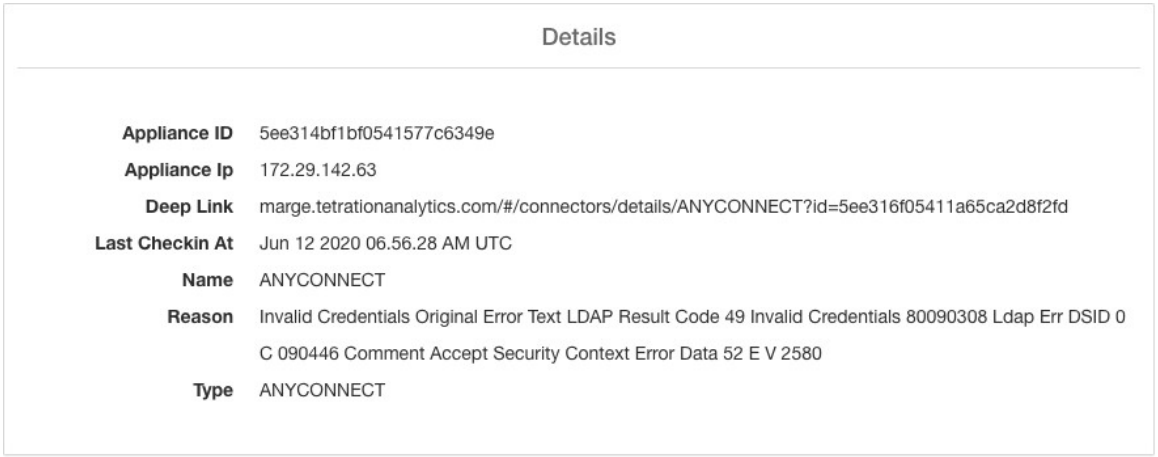


許可されている **Secure Workload** 仮想アプライアンス：Secure Workload Ingest および Secure Workload Edge

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 *111* : 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、「共通アラート構造」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"
}
```

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。