外部オーケストレータ

外部オーケストレータを使用して、ネットワーク上のシステムからワークロードに関する既存のメタデータを収集できます。一部の外部オーケストレータは、セグメンテーションポリシーを適用することもできます。

ワークロードラベルによるレコード承認システムが存在する環境の場合は、外部オーケストレータと連携してラベルを自動的にインポートする方法を用意しています。レコードシステムの変更は、Secure Workload によって自動的に学習され、インベントリのラベルを更新するために使用されます。ラベルの機能と用途の詳細については、「ワークロードラベル」を参照してください。

現在サポートされている外部オーケストレータ：

表 *1*：現在サポートされている外部オーケストレータ

タイプ	説明/用途
VMware vCenter	ホスト名、IP アドレス、ラベルなどの仮想マシンデータを vCenter サーバーから Secure Workload インポートする際に使用します。生成されたラベルを使用して、Secure Workload の範囲と適用ポリシーを作成できます。
Amazon Web Services	（新しい AWS オーケストレータを作成することはできません。代わりに、AWS コネクタを作成します。「AWS コネクタ」を参照してください。既存の AWS オーケストレータは読み取り専用です)。ホスト名、IP アドレス、ラベルなどの EC2 サーバーインスタンスのデータを特定の AWS アカウントから Secure Workload にインポートする際に使用します。生成されたラベルを使用して、Secure Workload の範囲とポリシーを作成できます。

タイプ	説明/用途
Kubernetes/OpenShift	ノード、ポッド、サービス、ラベルなどの Kubernetes のエンティティをインポートする際に使用します。これらのラベルを Secure Workload で使用して、範囲とポリシーを定義できます。
DNS	ゾーン転送により DNS サーバーから A/AAAA や CNAME レコードをインポートする際に使用します。これにより、Secure Workload の範囲とポリシーを定義する際に便利な DNS 名がラベルとして生成されます。
Infoblox	IPAM または DNS が有効になっている Infoblox アプライアンスから、拡張可能な属性を持つネットワーク、ホスト、および A/AAAA レコードをインポートする際に使用します。インポートされた拡張可能な属性は、Secure Workload の範囲およびポリシーのラベルとして使用できます。
F5 BIG-IP	特定の F5 ロードバランサから仮想サーバー構成を読み取り、提供サービスのラベルを生成します。このラベルは、Secure Workload で適用ポリシーを定義するために使用できます。ポリシー適用機能は、F5 REST API を介してそれらを F5 ポリシー規則に変換します。
Citrix Netscaler	特定の Netscaler ロードバランサから仮想サーバー構成を読み取り、提供サービスのラベルを生成します。このラベルは、Secure Workload で適用ポリシーを定義するために使用できます。ポリシー適用機能は、Netscaler の REST API を介してポリシーを Netscaler ACLに変換します。
Cisco Secure Firewall Management Center *(BETA)*	REST API を使用して、特定の Cisco Secure Firewall Management Center に登録されているすべての Cisco Secure Firewall Threat Defense（旧称 Firepower Threat Defense または FTD）デバイスにポリシーを展開する際に使用します。

- [外部オーケストレータ（External Orchestrators）] ページへの移動 （3 ページ）
- 外部オーケストレータの一覧表示 （3 ページ）
- 外部オーケストレータの作成 （4 ページ）

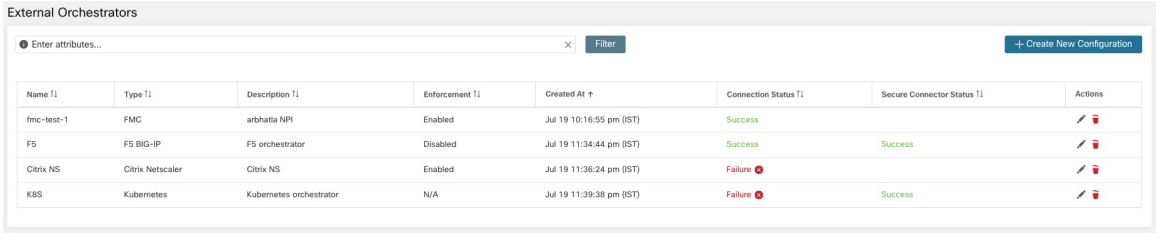
[外部オーケストレータ（External Orchestrators）] ページへの移動

外部オーケストレータのメインページには、左側のメニューバーから **[管理（Manage）]** > **[外部オーケストレータ（External Orchestrators）]** を選択してアクセスできます。

外部オーケストレータの一覧表示

外部オーケストレータのメインページには、既存の外部オーケストレータが表示されます。また、外部オーケストレータを変更および削除する機能と、新しい外部オーケストレータを作成する機能があります。

図 *1*: 外部オーケストレータのメインページ

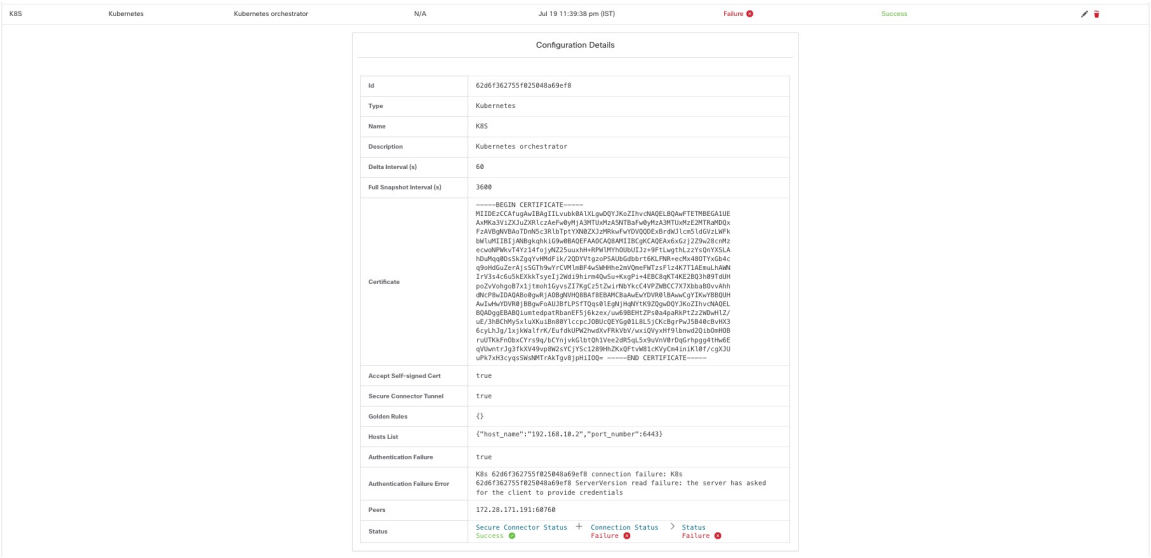


各行には、外部オーケストレータの短いバージョンが、[名前（Name）]、[タイプ（Type）]、[説明（Description）]、[適用（Enforcement）]、[作成日時（Created at）]、[接続ステータス（Connection Status）]、および [Secure Connectorステータス（Secure Connector Status）] とともに表示されます。[接続ステータス（Connection Status）] には、指定された外部データソースへの接続が成功したか失敗したかが示されます。[Secure Connectorステータス（Secure Connector

Status）] には、Secure Connector トンネルのステータス（[成功（Success）] または [失敗（Failure）]）が表示されます。トンネルが有効になっていない場合は、N/A が表示されます。

Secure Connector トンネルは、外部オーケストレータ構成の作成中に有効化できます。Secure Connector トンネルが有効になっている場合、外部オーケストレータの [接続ステータス（Connection Status）] は、認証ステータスと Secure Connector ステータスの両方に依存します。Secure Connector トンネルが有効になっていない場合、外部オーケストレータの [接続ステータス（Connection Status）] は認証ステータスのみに依存します。[成功（Success）] または [失敗（Failure）] のステータスに関係なく、それぞれの行をクリックして詳細を表示できます。Secure Connector クライアントメトリックの詳細を確認するには、[ステータス（Status）] 行をクリックするか、左ペインで [管理（Manage）] > [Secure Connector] に移動します。

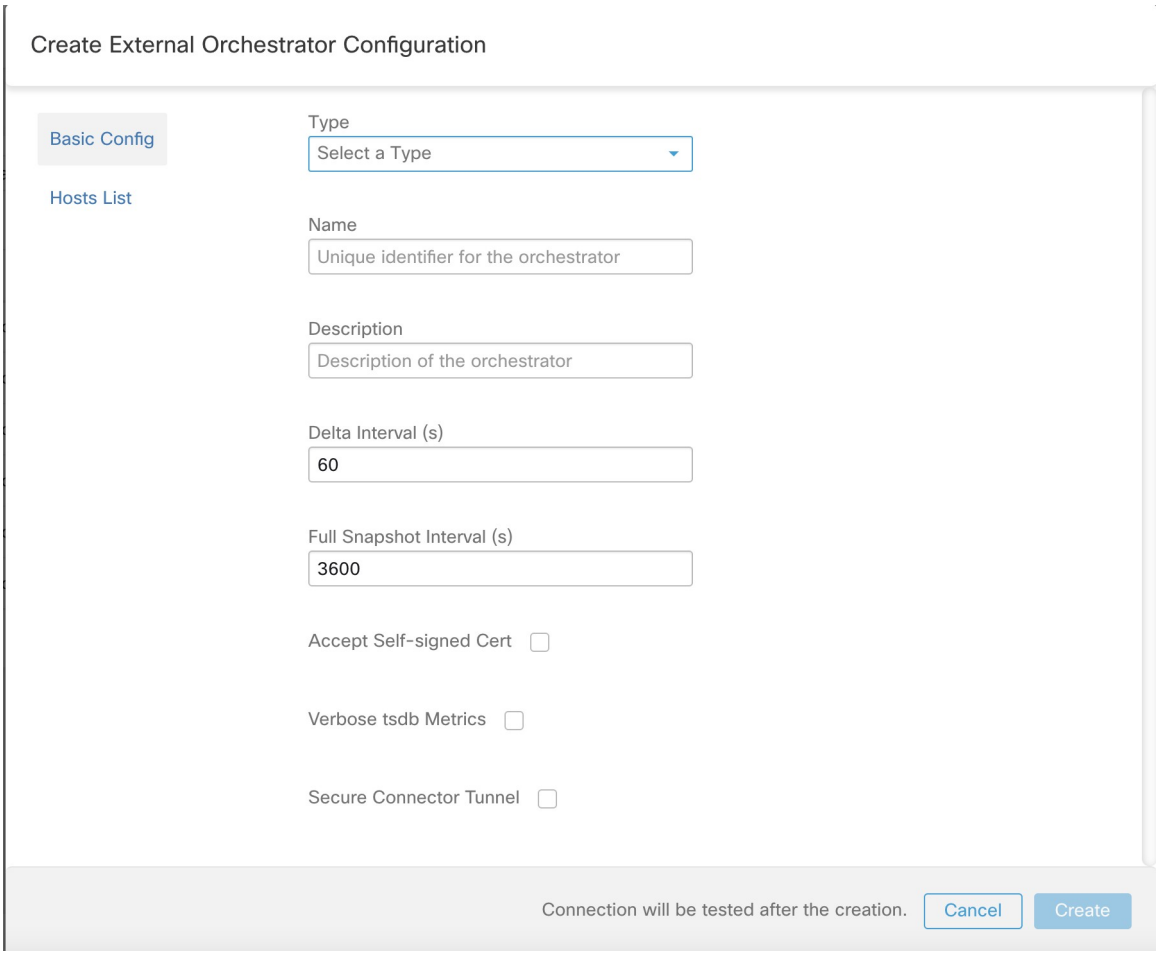
図 *2:* 外部オーケストレータ認証の失敗例



外部オーケストレータの作成

新しい外部オーケストレータを作成するには、外部オーケストレータのメインページで [新しい設定の作成（Create New Configuration）] ボタンをクリックします。これによりモーダルダイアログが表示され、名前を入力して外部オーケストレータタイプを選択できます。下の図は、基本設定ページを示しています。

図 *3* : 外部オーケストレータ設定の作成



次の表では、外部オーケストレータの共通フィールドについて説明します。選択したタイプに応じて、[基本設定（Basic Config）]ページで追加のパラメータを指定する必要があります。パラメータについては、以下に示す個々の外部オーケストレータのそれぞれのセクションで取り上げます。

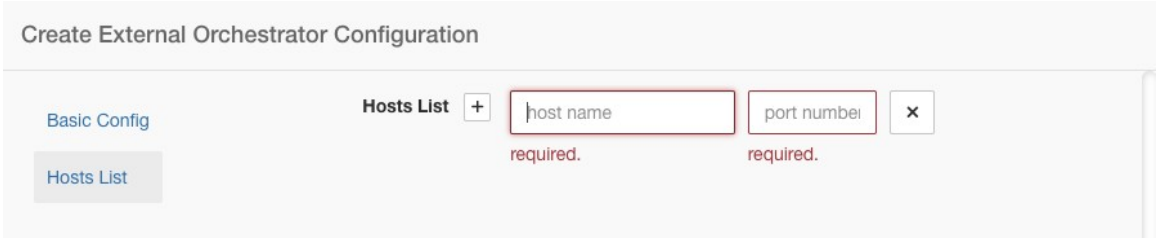
共通フィールド	必須	説明
タイプ（Type）	○	リストから外部オーケストレータを選択します。
名前（Name）	◯	アクティブなテナントに対して一意である必要がある外部オーケストレータの名前。
説明（Description）	×	外部オーケストレータの説明。

共通フィールド	必須	説明
フルスナップショット間隔（Full Snapshot Interval(s)）	○	外部オーケストレータが、選択した [タイプ（Type）] から設定の完全なスナップショットをインポートしようとする間隔（秒）。
自己署名証明書の受け入れ（Accept Self-signed Cert）	×	選択した [タイプ（Type）] から設定データを取得するために、Secure Workload で使用される HTTPS 接続の自己署名サーバー証明書を受け入れるには、このオプションをオンにします。デフォルトでは、自己署名サーバー証明書は許可されません。
セキュアコネクタトンネル（Secure Connector Tunnel）	×	Secure Workload クラスタへの接続が Cisco Secure Connector トンネルを介してトンネリングされるように設定するには、このオプションをオンにします。

（注） 上の図に示されている [デルタ間隔（Delta interval）] および [詳細なTSDBメトリック（Verbose TSDB Metrics）] フィールドはオプションであり、以下のそれぞれの説明で記載されている特定の外部オーケストレータのみが対象になります。

外部オーケストレータのタイプが [AWS] である場合を除き、[ホストリスト（Hosts List）] を指定する必要があります。外部オーケストレータがデータを取得してラベルを生成する外部データソースのネットワークアドレスを指定します。これを行うには、次の図に示すように、左側の [ホストリスト（Hosts List）] タブをクリックします。

図 *4:* 外部オーケストレータのホストリスト

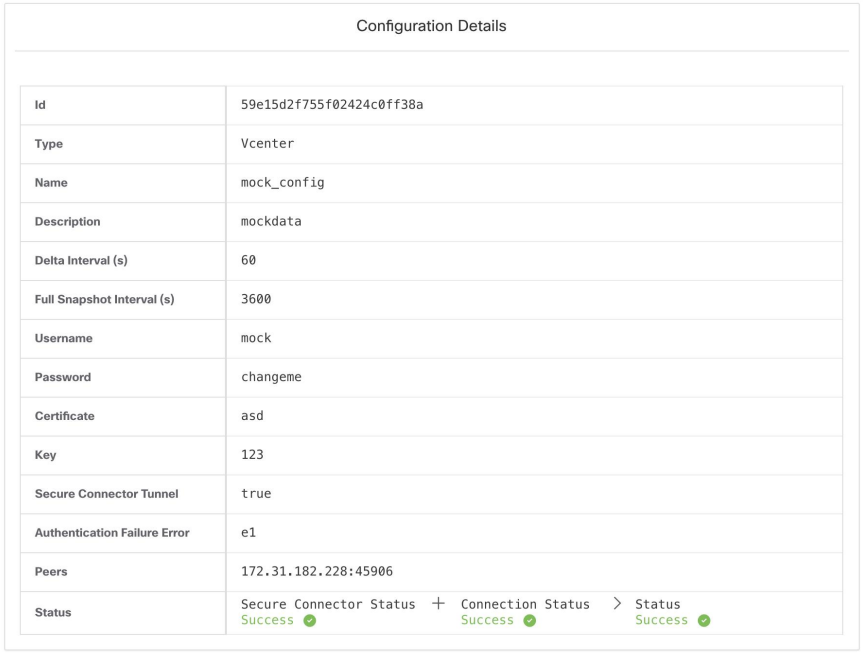


新しいホストリストエントリを追加するには、プラス記号をクリックします。各行には、有効なDNSホスト名、IPv4またはIPv6アドレス、およびポート番号が含まれている必要がありま

す。選択した外部オーケストレータタイプに応じて、高可用性または冗長性の目的で複数のホストを入力できます。詳細については、以下の選択した外部オーケストレータのそれぞれの説明を参照してください。

[作成（Create）] ボタンをクリックして新しい外部オーケストレータを作成します。リストビューの該当する行をクリックすると、設定の詳細を表示できます。

図 *5 :* 外部オーケストレータ設定の詳細

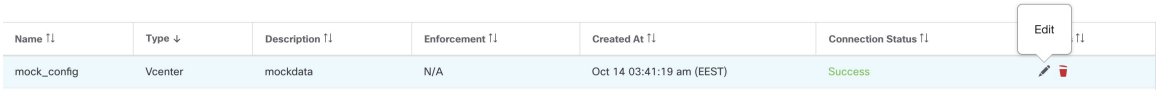


（注） 外部オーケストレータからの最初の完全スナップショットプルは非同期操作であるため、接続ステータスフィールドが更新されるまでに約 1 分かかります。

外部オーケストレータの編集

以下に示すように、外部オーケストレータの行の右側にある鉛筆ボタンをクリックして、構成を変更できる外部オーケストレータを作成する場合と同様のモーダルダイアログを開きます。

図 *6 :* 外部オーケストレータの編集



（注）

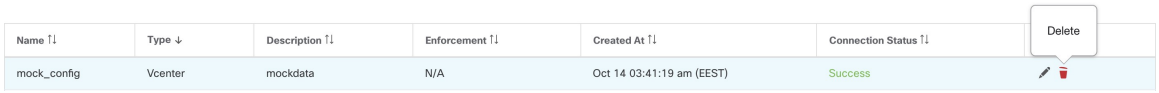
- [タイプ（Type）] フィールドは編集できません。
- 認証にキーと証明書を使用する構成の場合、構成を更新するたびにキーと証明書を提供する必要があります。
- 外部オーケストレータの構成変更は非同期操作であるため、[接続ステータス（Connection Status）] フィールドが更新され、入力された変更が正しいことが確認されるまでに約 1 分かかります。

[更新（Update）] ボタンをクリックして、構成に加えた変更を保存します。

外部オーケストレータの削除

注意 外部オーケストレータを削除すると、そのオーケストレータによって提供されたラベルも削除され、ポリシーに影響します。外部オーケストレータを削除するには、以下に示すようにごみ箱ボタンをクリックします。

図 7: 外部オーケストレータの削除



オーケストレータにより生成されるラベル

Cisco Secure Workload は、次のラベルをすべての AWS インスタンスに追加します。

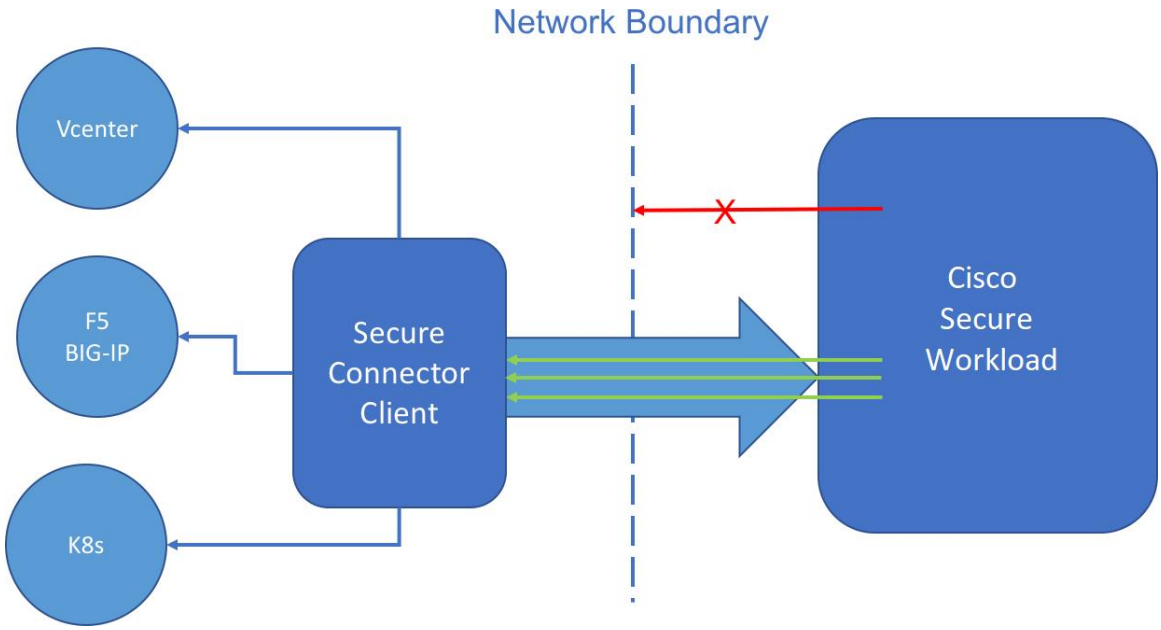
キー	値
orchestrator_system/orch_type	aws
orchestrator_system/cluster_name	<*Cluster_name* はユーザーがこのオーケストレータの設定に付けた名前>
orchestrator_system/cluster_id	<|*product*| でのオーケストレータの設定の *UUID*>

Secure Connector

Secure Workload を使用して、ユーザータグをインポートしたり、外部オーケストレータ（「外部オーケストレータ」を参照）にポリシーを適用したりするには、Secure Workload がオーケス

トレータ API サーバー（vCenter、Kubernetes、F5 BIG-IP など）への発信接続を確立する必要があります。Secure Workload クラスタからオーケストレータへの直接着信接続を許可できない場合があります。Secure Connector は、オーケストレータと同じネットワークから Secure Workload クラスタへの発信接続を確立することにより、この問題を解決します。この接続は、クラスタからの要求をオーケストレータ API サーバーに渡すためのリバーストンネルとして使用されます。

図 *8 : Secure Connector*



ルート範囲ごとに、一度にアクティブにできるトンネルは1つだけです。追加のトンネルを開始しようとすると、1 つのトンネルがすでにアクティブであることを示すエラーメッセージが表示されて拒否されます。アクティブトンネルを使用して、クライアントが実行されているネットワークから到達可能な複数のオーケストレータに接続できます。オーケストレータごとの構成は、そのオーケストレータへの接続が Secure Connector トンネルを経由する必要があるかどうかを示すために使用されます。

Secure Connector クライアントと Secure Workload クラスタ間の通信はすべて相互に認証され、TLS を使用して暗号化されます。

セキュリティを向上させるために、適切に保護されている隔離されたマシンに Secure Connector クライアントをインストールすることを推奨します。そのようなマシンには、Secure Workload クラスタへの発信接続のみを許可するファイアウォールルールが必要であり、外部のオーケストレータ API サーバー Secure Workload にはアクセスを許可する必要があります。

Secure Connector トンネルを使用するようにオーケストレータを構成するには、製品の外部オーケストレータを構成する手順を参照してください。

Secure Connector の OpenAPI エンドポイントの詳細については、「Secure Connector API エンドポイント」を参照してください。

技術的な詳細情報

トンネルをブートストラップするために、Secure Connector クライアントが公開キーと秘密キーのペアを作成し、サーバーがリモートで公開キーの証明書に署名します。このリモート署名プロセスを保護し、クライアントが属するルート範囲を識別するために、暗号化された 1 回限りの期間限定トークンが使用されます。サーバー側ではルート範囲ごとに一意の証明書があります。クライアントはこの証明書をサーバーの認証に使用します。証明書は定期的にローテーションされ、通信の機密性が維持されます。

Secure Connector クライアントは、トンネルクライアントと SOCKS5 サーバー内に構築されます。トンネルが開始されると、クライアントは Secure Workload クラスタからのトンネル着信接続を待ちます。着信接続は SOCKS5 サーバーによって処理され、宛先ホストに転送されます。

Secure Connector の要件

Secure Connector クライアントの要件：

- RHEL/CentOS 7（x86_64）
- 2 CPU コアと 4 GB RAM
- Secure Connector を使用するオンプレミス オーケストレータからのデータを処理するのに十分なネットワーク帯域幅
- ポート 443 での Secure Workload クラスタへの発信接続（直接または HTTP(S) プロキシ経由）
- 内部オーケストレータ API サーバーへの発信接続（直接）

Secure Connector クライアントの導入

プロキシ サポート

Secure Connector クライアントは、HTTP(S) プロキシを介した Secure Workload クラスタへの接続をサポートしています。必要に応じて、クライアントの HTTPS_PROXY 環境変数を設定して、プロキシサーバーを設定する必要があります。変数を設定するには、/etc/systemd/system/tetration-secure-connector.service にある systemd サービスファイルの [Service] セクションに次の行を追加します。この設定は、再インストール後は維持されません。スティッキ設定の場合、この行は /etc/systemd/system/tetration-secure-connector.service.d/10-https-proxy.conf の新しいファイルに追加できます。いずれかの設定を有効にするには、*systemctl daemon-reload* を実行して systemd 設定をリロードします。

```
[Service]
Environment="HTTPS_PROXY=<Proxy Server Address>"
```

展開の概要

Secure Connector は、オーケストレータ API サーバーに到達するために、Secure Workload クラスタから内部ネットワークへのリバーストンネルを作成します。

Secure Connector クライアントを起動するには、Secure Connector RPM をダウンロードして、1 回限りの登録トークンを生成する必要があります。

1. サポートされているプラットフォームに 最新の Secure Connector クライアント RPM のダウンロード。
2. 登録トークンの生成。
3. トークンをコピーしてクライアントを開始する、クライアントを起動します。

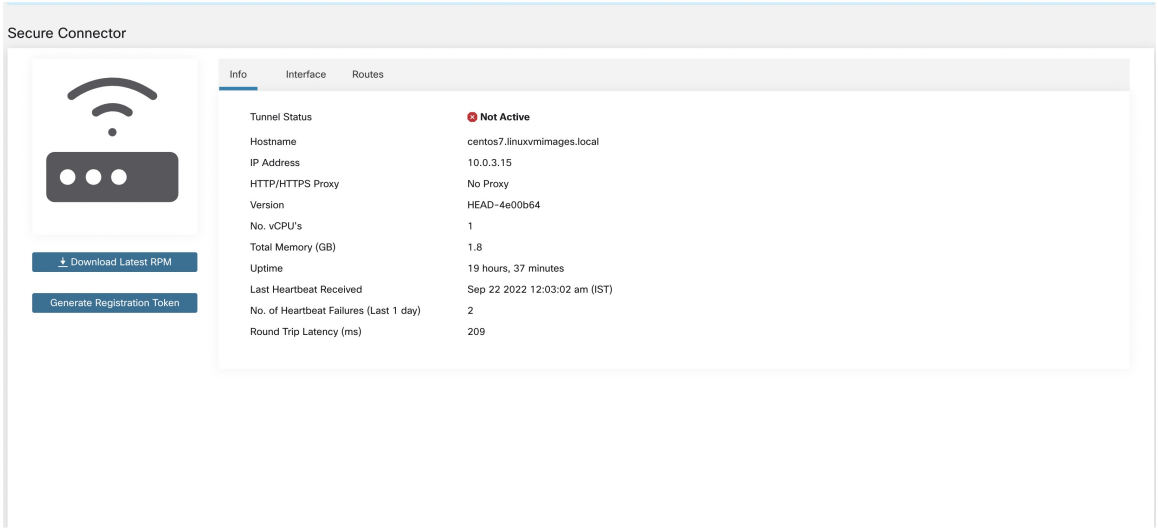
Secure Connector クライアントの導入

最新の Secure Connector クライアント RPM のダウンロード

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）] > [ワークロード（Workloads）] > [セキュアコネクタ（Secure Connector）]** をクリックします。

ステップ 2 [最新のRPMをダウンロード（Download Latest RPM）] をクリックします。

図 *9 : [セキュアコネクタ（Secure Connector）]* ページ



ステップ 3 RPM パッケージを展開用の Linux ホストにコピーし、ルート権限で次のコマンドを実行します。

```
rpm -ivh <rpm_filename>
```

登録トークンの生成

ステップ 1 **[管理（Manage）]** > **[ワークロード（Workloads）]** > **[セキュアコネクタ（Secure Connector）** の順にクリックします。

ステップ 2 [登録トークンの生成（Generate Registration Token）] をクリックします。

トークンをコピーしてクライアントを開始する

Secure Connector ページで登録トークンを生成すると、クライアントをブートストラップするための 1 回限りの期間限定トークンを含む *registration.token* ファイルが作成されます。このトークンファイルは、Secure Connector クライアントパッケージをインストールしたホストにコピーする必要があります。トークンファイルをコピーする前に、ホスト上で Secure Connector クライアントが停止していることを確認してください。次のコマンドを使用して、クライアントを停止できます。

```
systemctl stop tetration-secure-connectorsystemctl stop tetration-secure-connector
```

トークンファイルを *cert* フォルダにコピーします。

```
/etc/tetration/cert/registration.token
```

Secure Connector クライアントを再起動します。

```
systemctl start tetration-secure-connector
```

（オプション）特定のバージョンの Secure Connector クライアントを展開する

ステップ 1 **特定のバージョンの Secure Connector クライアント RPM のダウンロード**

a) 左側のナビゲーションバーで、**[管理（Manage）]** > **[ワークロード（Workloads）]** > **[エージェント（Agents）]** をクリックします。

b) [インストーラ（Installer）] タブをクリックします。

c) [クラシックパッケージインストーラを使用した手動インストール（Manual Install using classic packaged installers）] を選択し、[次へ（Next）] をクリックします。

Secure Connector クライアントパッケージのエージェントタイプは「Secure Connector」です。

d) 適切なバージョンを見つけ（クラスタに複数ある場合）、[ダウンロード（Download）] をクリックします。

e) RPM パッケージを展開用の Linux ホストにコピーし、ルート権限で次のコマンドを実行します。`rpm -ivh<rpm_filename>`

ステップ 2 **API を使用した新しいトークンの取得**

Secure Connector トークンは、OpenAPI（Get Tokenendpoint）を使用して取得することもできます。次の Python および Bash スニペットを使用して、新しいトークンを取得できます。使用される API キーには *external_integration* 機能が必要であり、指定されたルート範囲への書き込みアクセスが必要であることに注意してください。Python 用の Secure Workload OpenAPI クライアントのインストールと新しい API キーの作成については、「OpenAPI 認証」を参照してください。

- トークン取得のための Python スニペット

```
from tetpyclient import RestClient from urllib import quote
API_ENDPOINT = "https://<UI_VIP_OR_DNS_FOR_TETRATION_DASHBOARD>"
ROOT_SCOPE_NAME = r"""<ROOT_SCOPE_NAME>"""
API_CREDENTIALS_FILE = "<API_CREDENTIALS_JSON_FILE>"
OUTPUT_TOKEN_FILE = "registration.token"
if __name__ == "__main__":
      client = RestClient(API_ENDPOINT,
                          credentials_file=API_CREDENTIALS_FILE) # Add (verify=False) to
‹→skip certificate verification
    escaped_root_scope_name = quote(ROOT_SCOPE_NAME, safe='')
    resp = client.get('/secureconnector/name/{}/token'.format(escaped_root_scope_name))
    if resp.status_code != 200:
      print 'Error ({}): {}'.format(resp.status_code, resp.content)
      exit(1)
else:
    with open(OUTPUT_TOKEN_FILE, 'w') as f:
     f.write(resp.content)
```

- トークン取得のための BASH スニペット

```
#!/bin/bash
HOST="https://<UI_VIP_OR_DNS_FOR_TETRATION_DASHBOARD>"
API_KEY="<API_KEY>"
API_SECRET="<API_SECRET>"
ROOTSCOPE_NAME="<ROOT_SCOPE_NAME>" # if the name contains spaces or special
‹→characters, it should be url-encodedTOKEN_FILE="registration.token"
TOKEN_FILE="registration.token"
INSECURE=1 # Set to 0 if you want curl to verify the identity of the cluster
METHOD="GET"
URI="/openapi/v1/secureconnector/name/$ROOTSCOPE_NAME/token"
CHK_SUM=""
CONTENT_TYPE=""
TS=$(date -u "+%Y-%m-%dT%H:%M:%S+0000")
CURL_ARGS="-v"
if [ $INSECURE -eq 1 ]; then
    CURL_ARGS=$CURL_ARGS" -k"
fi
MSG=$(echo -n -e "$METHOD\n$URI\n$CHK_SUM\n$CONTENT_TYPE\n$TS\n")
SIG=$(echo "$MSG"| openssl dgst -sha256 -hmac $API_SECRET -binary | openssl enc -
‹ →base64)
REQ=$(echo -n "curl $CURL_ARGS $HOST$URI -w '%{http_code}' -H 'Timestamp: $TS' -H
‹ →'Id: $API_KEY' -H 'Authorization: $SIG' -o $TOKEN_FILE")
status_code=$(sh -c "$REQ")
if [ $status_code -ne 200 ]; then
    echo "Failed to get token. Status: " $status_code
else
    echo "Token retrieved successfully"
fi
```

ステップ 3　トークンをコピーしてクライアントを開始する

```
systemctl stop tetration-secure-connectorsystemctl stop tetration-secure-connector
```

トークンファイルを *cert* フォルダにコピーします。

```
/etc/tetration/cert/registration.token
```

Secure Connector クライアントを再起動します。

```
systemctl start tetration-secure-connector
```

Secure Connector クライアントの状態の確認

rpmdb でパッケージ tet-secureconnector-client-site をクエリすることで、Secure Connector クライアントがインストールされているか確認できます。

```
rpm -q tet-secureconnector-client-siterpm -q tet-secureconnector-client-site
```

インストールされているクライアントの現在の状態を確認するには、systemd サービス tetration-secure- connector のステータスを確認します。

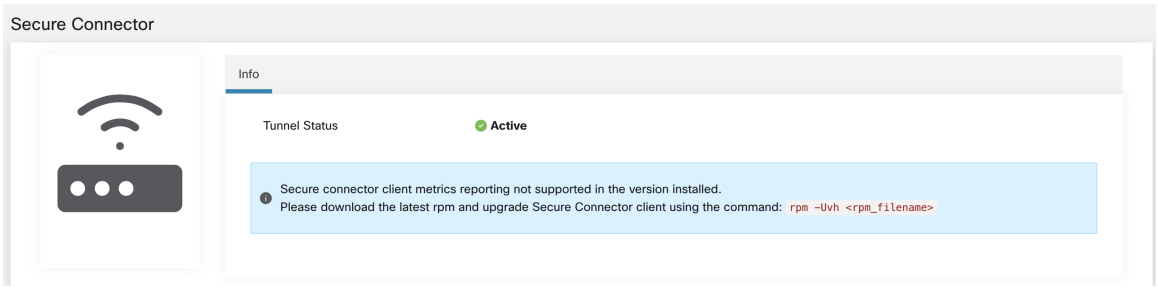
```
systemctl status tetration-secure-connector
```

Secure Connector クライアントステータス

[外部オーケストレータ（External Orchestrators）] ページに、設定された外部オーケストレータと Secure Connector トンネルのステータスが表示されます。外部オーケストレータの設定中に Secure Connector が有効になっている場合、[Secure Connector] ページで Secure Connector クライアントメトリックを表示できます。

ただし、Secure Connector トンネルのステータスが [アクティブ（Active）] であるにもかかわらず、クライアントメトリックが表示されない場合は、古いバージョンの Secure Connector がインストールされていることを意味します。Secure Connector クライアントバージョンのアップグレードに関する、次のようなメッセージが表示されます。

図 *10 : Secure Connector* クライアントのアップグレードメッセージ



（注） 最新の Secure Connector RPM のインストール手順については、「最新の Secure Connector クライアント RPM のダウンロード」を参照してください。

クライアントメトリックを表示するには、次の手順を実行します。

ステップ 1 [設定の詳細（Configure Details）] で、[ステータス（Status）] 行をクリックします。[Secure Connector] ページが表示されます。

（注） 左ペインで**[管理（Manage）]** > **[Secure Connector]**を選択して、Secure Connector トンネルのステータスにアクセスすることもできます。

ステップ 2 [全般（General）]、[インターフェイス（Interface）]、または [ルート（Routes）] タブを選択して、クライアントと Secure Workload クラスタ間の接続ステータスの詳細にアクセスします。

タブ	説明
General	次の情報を一覧表示します。 • Tunnel Status • ホスト名 • IP アドレス • HTTP/HTTPS プロキシ（HTTP/HTTPS Proxy） • [バージョン（Version）]：ビルドバージョンを一覧表示します。 • vCPU の数 • 合計メモ（GB） • [稼働時間（Uptime）]：Secure Connector クライアントが実行されている VM の稼働時間を一覧表示します。 • [ハートビートの最終受信日時（Last Heartbeat Received）]：クライアントから最後に受信したハートビートの日付とタイムスタンプを一覧表示します。 • [ハートビートの失敗回数（過去 1 日）（No. of Heartbeat Failures (Last 1 day)）]：Secure Connector クライアントへの接続が 1 日に失敗した回数をリストします。クライアントが非アクティブのままである場合、カウントは増えませません。カウントは 1 日の終わりにリセットされます。 • ラウンドトリップ遅延（ミリ秒）
インターフェイス	Secure Connector クライアントが実行されている VM のインターフェイスの詳細を一覧表示します。
ルート	ルートテーブルには、宛先 IP アドレス、ゲートウェイ、genmask、およびインターフェイスが一覧表示されます。

Secure Connector クライアントのアップグレード

Secure Connector クライアントは自動更新をサポートしていません。ソフトウェアの新しいバージョンをインストールするには、次のコマンドを使用して現在のバージョンをアンインストールしてから、新しいバージョンの（インストール手順）に進みます。

rpm -e tet-secureconnector-client-siterpm -e tet-secureconnector-client-site

Secure Connector クライアントの削除

Secure Connector クライアントは、次のコマンドを使用してアンインストールできます。

rpm -e tet-secureconnector-client-siterpm -e tet-secureconnector-client-site

Amazon Web Services

（注） AWS 外部オーケストレータ機能は、新しい AWS クラウドコネクタ機能の一部になりました。このリリースにアップグレードすると、既存の AWS 外部オーケストレータは読み取り専用になります。変更が必要な場合は、新しい AWS コネクタを作成します。詳細については、「AWS コネクタ」を参照してください。

Secure Workload では AWS リージョンからリアルタイムでインベントリデータを自動的に取り込むことができます。「AWS」タイプの外部オーケストレータ設定が追加されると、Secure Workload アプライアンスは AWS エンドポイントに接続し、実行中および停止状態のすべてのインスタンスのメタデータを取得します。

前提条件

- 使用されるセキュリティトークン（アクセスキーと秘密鍵）には、オーケストレータ情報の取得を許可する適切な種類の IAM 権限が必要です。

設定フィールド

属性	説明
ID	オーケストレーションの固有識別子。
名前	ユーザーが指定したオーケストレータの名前。
タイプ	オーケストレータのタイプ：（この場合は aws）
説明	オーケストレータの簡単な説明。

属性	説明
AWS アクセス キー ID	オーケストレータ構成が作成されているアカウントに関連付けられているアクセスキー。
AWS シークレットアクセスキー	オーケストレータ構成が作成されているアカウントに関連付けられている秘密鍵。構成を編集するたびに秘密鍵を再入力する必要があることに注意してください。
AWS リージョン	ワークロードが展開されているリージョン。ワークロードが複数のリージョンに分散している場合は、リージョンごとに個別の構成が必要です。正しいリージョン値については、https://docs.aws.amazon.com/general/latest/gr/rande.htmlを参照してください。
自己署名証明書の受け入れ（Accept Self-signed Cert）	AWS の場合、自動的に true とマークされます。ユーザーは編集できません。
フルスナップショット間隔	秒単位のフルスナップショット間隔。Orchestrator Inventory Manager は、オーケストレータからフル更新ポーリングを実行します。
差分スナップショット間隔	秒単位の差分スナップショット間隔。Orchestrator Inventory Manager は、オーケストレータから増分更新のみをフェッチします。
ホストリスト	AWS オーケストレータタイプにはホストリストは必要ありません。AWS のエンドポイントは、前述の AWS リージョンフィールドから取得されます。このフィールドは空白のままにする必要があります。
詳細な TSDB メトリック	有効にすると、個々のオーケストレータの TSDB メトリックがレポートされます。無効な場合、すべてのオーケストレータメトリックの集計がレポートされます。
セキュアコネクタトンネル（Secure Connector Tunnel）	Secure Connector を介したこのオーケストレータのホストへのトンネル接続 します。

ワークフロー

- 前述の構成フィールドを入力した AWS オーケストレータを構成します。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、次のラベルをすべての AWS インスタンスに追加します。

キー	値
orchestrator_system/orch_type	aws
orchestrator_system/cluster_name	<*Cluster_name* はユーザーがこのオーケストレータの設定に付けた名前>
orchestrator_system/cluster_id	<|*product*| でのオーケストレータの設定の *UUID*>

インスタンス固有のラベル

次のラベルはインスタンス固有です。

キー	値
orchestrator_system/workload_type	vm
orchestrator_system/machine_id	<*AWS* によって割り当てられた *InstanceID*>
orchestrator_system/machine_name	<*AWS* によってこのノードに指定された *PublicDNS (FQDN)*>
orchestrator_‘<AWS Tag Key>‘	<*AWS* タグ値>

トラブルシューティング

- AWS リージョンと可用性ゾーンを混同する。

  これらの値は両方とも相互に関連しているため、混同しないでください。たとえば、us-west-1 がリージョンの場合、可用性ゾーンは us-west-1a、us-west-1b などになります。オーケストレータを構成するときは、リージョンを使用する必要があります。すべてのリージョンについては、https://docs.aws.amazon.com/general/latest/gr/rande.html [英語] を参照してください。

- オーケストレータ構成を更新した後の接続およびクレデンシャルの問題。

  ユーザーは、設定が更新されるたびに *AWS* シークレットキーを再送信する必要があります。

Kubernetes/OpenShift

（注） EKS および AKS 外部オーケストレータ機能は、それぞれ新しい AWS および Azure クラウドコネクタ機能の一部になりました。このリリースにアップグレードすると、既存の EKS および AKS 外部オーケストレータは読み取り専用になります。変更が必要な場合は、新しい AWS または Azure コネクタを作成します。詳細については、「クラウドコネクタ」の関連トピックを参照してください。

シンプルな Kubernetes および OpenShift の外部オーケストレータは変更されていません。

Cisco Secure Workload は、Kubernetes クラスタからライブのインベントリの自動取り込みをサポートします。Kubernetes/OpenShift クラスタに外部オーケストレータ構成が追加されると、Secure Workload はクラスタの API サーバーに接続し、そのクラスタ内のノード、ポッド、およびサービスのステータスを追跡します。Secure Workload は、オブジェクトタイプごとに、オブジェクトに関連付けられているすべての Kubernetes ラベルをインポートします。値はすべてそのままインポートされます。

Secure Workload は、Kubernetes/OpenShift オブジェクト用に定義されたラベルのインポートに加えて、それらのオブジェクトをインベントリフィルタで使用しやすくするラベルも生成します。それらの追加ラベルは、範囲とポリシーを定義する際に特に役立ちます。

すべてのラベルの詳細については、「Kubernetes クラスタに関連するラベル」を参照してください。

Kubernetes ノードで適用が有効になっている場合（適用エージェントがインストールされ、構成プロファイルで適用エージェントでの適用が有効になっている）、適用ポリシーは、この統合を介して取り込まれた Kubernetes エンティティに関する情報を使用して、ノードおよびポッド名前空間の内部の両方にインストールされます。

クラウドプラットフォーム上の Kubernetes について

サポートされているクラウドプラットフォームで実行されている次のマネージド Kubernetes サービスの場合、このオーケストレータの機能は、クラウドコネクタを使用して提供されます。

- Amazon Web Services（AWS）で実行される Elastic Kubernetes Service（EKS）
- Microsoft Azure で実行される Azure Kubernetes Service（AKS）
- Google Cloud Platform（GCP）で実行される Google Kubernetes Engine（GKE）

クラウドプラットフォーム上の Kubernetes クラスタからデータを取得する方法の詳細については、「クラウドコネクタ」のトピックを参照してください。

要件および前提条件

- サポートされている Kubernetes および OpenShift バージョンについては、https://www.cisco.com/go/secure-workload/requirements/integrationsを参照してください。
- Secure Connector トンネル（接続に必要な場合）。

設定フィールド

次の設定フィールドは、オーケストレータオブジェクトの Kubernetes オーケストレータ設定に関連します。

フィールド	説明
[名前（Name）]	ユーザーが指定したオーケストレーションの名前。
[説明（Description）]	オーケストレーションのユーザー指定の説明。
[デルタ間隔（Delta Interval）]	Kubernetes エンドポイントの変更を確認する間隔（秒）
[フルスナップショット間隔（Full Snapshot Interval）]	Kubernetes データの完全なスナップショットを実行する間隔（秒単位）
[ユーザー名（Username）]	オーケストレーション エンドポイントのユーザー名。
[パスワード（Password）]	オーケストレーション エンドポイントのパスワード。
[証明書（Certificate）]	認証に使用されるクライアント証明書
[キー（Key）]	クライアント証明書に対応するキー
[認証トークン（Auth Token）]	Opaque 認証トークン（ベアラートークン）。
[CA証明書（CA Certificate）]	オーケストレーション エンドポイントを検証する CA 証明書
[自己署名証明書の受け入れ（Accept Self-signed Cert）]	Kubernetes API サーバー証明書の厳密な SSL チェックを無効にするチェックボックス
[詳細なTSDBメトリック（Verbose TSDB Metrics）	Kubernete オーケストレータのメトリックごとに維持されます。False に設定されている場合、Secure Workload クラスタ全体のメトリックのみが維持されます。

フィールド	説明
[セキュアコネクタトンネル（Secureconnector Tunnel）]	このオーケストレータのホストへのセキュアコネクタトンネルを介したトンネル接続
[ポリシー検出のクラスタリングに使用（Use for policy discovery clustering）]	自動ポリシー検出時にクラスタを作成する場合、このKubernetesクラスタのラベルメタデータを含めます。これにより、レプリカセットや環境などの追加のリソース情報も収集され、より正確なクラスタが生成されます。現在のところ、このオプションはvanilla KubernetesおよびOpenShiftクラスタでのみ使用できます。GKE、AKS、EKSなどのクラウドベースのKubernetesサービスはまだサポートされていません。
[ホストリスト（Hosts List）]	オーケストレータへのSecure Workloadの接続方法を指定する{“host_name”, port_number}ペアの配列
[K8sマネージャータイプ（K8s manager type）]	Kubernetesクラスタのマネージャタイプ（Vanilla/Openshift Kubernetesの環境以外）
[AWSクラスタ名（AWS cluster name）]	クラスタの作成時に指定されたオーケストレータの名前（既存のEKSのみ）
[AWSアクセスID（AWS Access ID）]	オーケストレータ構成が作成されているアカウントに関連付けられているアクセスキー（既存のEKSのみ）
[AWS秘密アクセスキー（AWS Secret Access Key）]	オーケストレータ構成が作成されているアカウントに関連付けられている秘密鍵。構成を編集するたびに秘密鍵を再入力する必要があることに注意してください（既存のEKSのみ）。
[AWSリージョン（AWS Region）]	ワークロードが展開されているリージョン。ワークロードが複数のリージョンに分散している場合は、リージョンごとに個別の設定が必要です。正しいリージョン値については、「https://docs.aws.amazon.com/general/latest/gr/rande.html」を参照してください（既存のEKSのみ）。

フィールド	説明
[AWS引継ぎロールのARN（AWS Assume Role ARN）]	オーケストレータへの接続中に引き継ぐロールの Amazon リソース番号。「https://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html」を参照してください（既存の EKS のみ）。
[AzureテナントID（Azure Tenant ID）]	Azure サブスクリプションに関連付けられたテナント ID （既存の AKS のみ）。
[AzureクライアントID（Azure Client ID）]	Azure AD で認証する必要があるアプリケーションに関連付けられたグローバルに一意の ID （既存の AKS のみ）。
[Azureクライアントシークレット（Azure Client Secret）]	Azure AD で認証する必要があるアプリケーションのサービスプリンシパルに関連付けられたパスワード（既存の AKS のみ）。

オーケストレータのゴールデンルール

ゴールデンルールオブジェクトの属性については、以下を参照してください。ゴールデンルールを使用することで、Kubernetes クラスタノードで適用が有効になった後も、Kubernetes クラスタが機能し続けるために必要なルールの仕様を簡潔に作成できます。

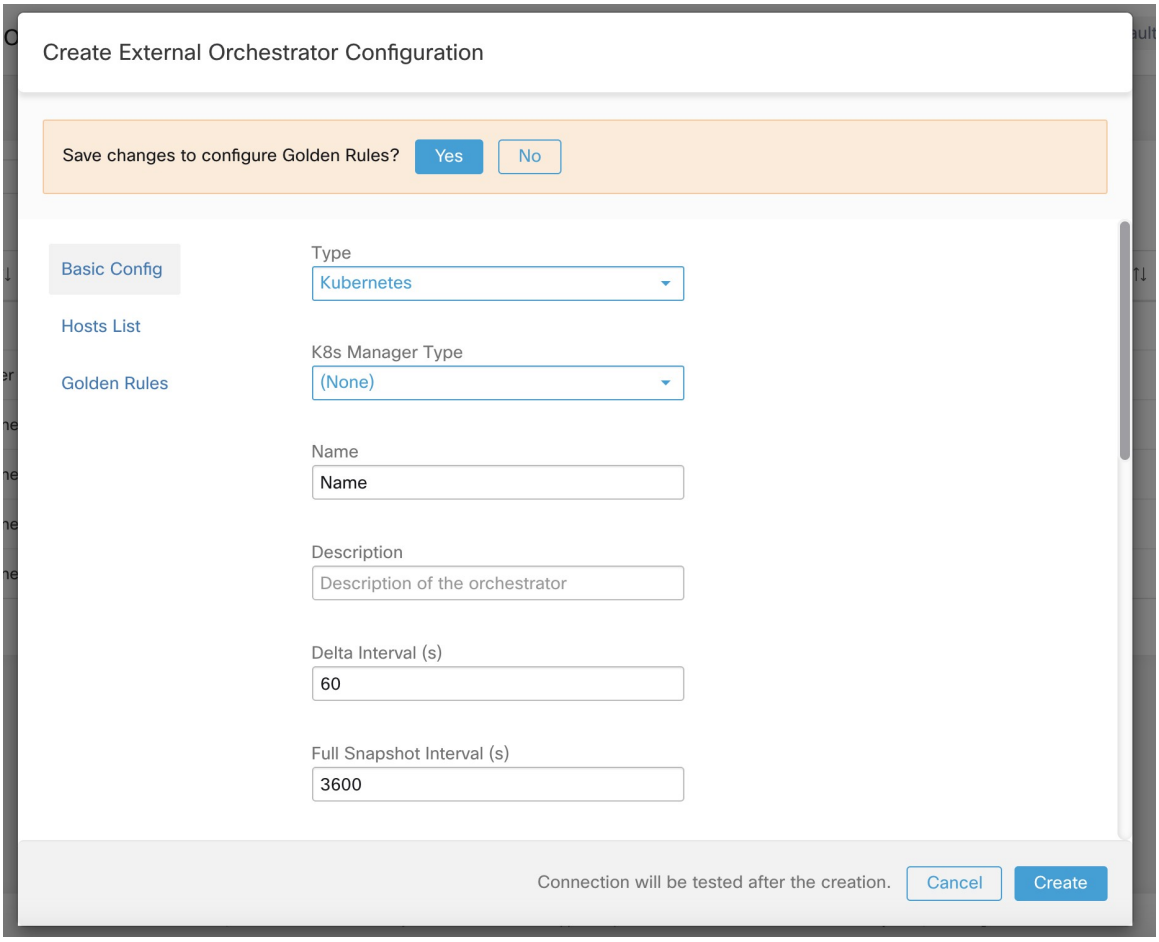
属性	説明
Kubelet ポート	Kubelet ノードのローカル API ポート
サービス	Kubernetes サービスオブジェクトの配列

kubelet ポートは、Kubernetes 管理デーモンから kubelet へのトラフィック（ライブログ、インタラクティブモードのポッドの実行など）を許可するポリシーを作成するために必要です。さまざまなKubernetesサービスとデーモン間の重要な接続は、一連のサービスとして指定され、各サービス配列のエントリには次の構造があります。

- 説明：サービスを説明する文字列
- アドレス：<IP>:<port>/<protocol> 形式のサービス エンドポイント アドレスのリスト。
- コンシューマ：エンドポイントのコンシューマのリスト（許可される値はポッドまたはノード）。

（注） タイプとして [Kubernetes] が選択されている場合、ゴールデンルール構成が許可されます。

図 *11 : Kubernetes* タイプのゴールデンルール構成の作成



ワークフロー

- 必要に応じて、Secure Workload クラスタから Kubernetes API サーバー（複数可）への接続用に Secure Connector トンネルを構成します。
- 前述の構成フィールドを入力した Kubernetes オーケストレータを構成します。
- Kubernetes オーケストレータのゴールデンルールを設定します。

Kubernetes ロールベース アクセス コントロール（RBAC）リソースに関する考慮事項

Kubernetes クライアントは、次のリソースを GET/LIST/WATCH しようとします。管理キー/証明書または管理サービスアカウントを構成しないことを強くお勧めします。

提供されるKubernetes認証資格情報には、次のリソースに対する最小限の権限セットが必要です。

リソース	Kubernetes の動詞
endpoints	[get list watch]
namespaces	[get list watch]
ノード	[get list watch]
pods	[get list watch]
サービス	[get list watch]
ingresses	[get list watch]
replicationcontrollers	[get list watch]
replicasets	[get list watch]
導入	[get list watch]
daemonsets	[get list watch]
statefulsets	[get list watch]
雇用	[get list watch]
cronjobs	[get list watch]

基本的に、これらの最小限の権限を使用して、Kubernetesサーバーに特別なサービスアカウントを作成できます。このサービスアカウントの作成を容易にするkubectlコマンドのシーケンスの例を以下に示します。clusterrole（roleではない）とclusterrolebindings（rolebindingsではない）を使用することに注意してください。これらはクラスタ全体のロールであり、名前空間ごとのロールではありません。role/rolebindingの使用は、機能しません。これは、Secure Workloadがすべての名前空間からのデータ取得を試行するためです。

```
$ kubectl create serviceaccount csw.read.only
```

clusterroleを作成します。

最小限の権限を持つclusterrole.yamlの例を以下に示します。

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
name: csw.read.only
rules:
- apiGroups:
- ""
resources:
- nodes
- services
- endpoints
- namespaces
- pods
```

```
- replicationcontrollers
- ingresses
verbs:
- get
- list
- watch
- apiGroups:
- extensions
- networking.k8s.io
resources:
- ingresses
verbs:
- get
- list
- watch
- apiGroups:
- apps
resources:
- replicasets
- deployments
- statefulsets
- daemonsets
verbs:
- get
- list
- watch
- apiGroups:
- batch
resources:
- jobs
- cronjobs
verbs:
- get
- list
- watch
$ kubectl create -f clusterrole.yaml
```

（注） これらのさまざまなリソースの API グループは、Kubernetes のバージョン間の変更の影響を受ける場合があります。上記のサンプルは Kubernetes バージョン 1.20 ～ 1.24 で動作しますが、他のバージョンではいくつかの調整が必要になる場合があります。

clusterrolebinding を作成します。

```
$ kubectl create clusterrolebinding csw.read.only --clusterrole=csw.read.
‹→only --serviceaccount=default:csw.read.only
```

serviceaccount から認証トークンシークレットを取得し（GUI の [認証トークン（Auth Token）] フィールドで使用）、base64 からデコードするために、yaml 出力で serviceaccount をリストして、シークレットの名前を取得できます。

```
$ kubectl get serviceaccount -o yaml csw.read.only
apiVersion: v1
kind: ServiceAccount
metadata:
creationTimestamp: 2020-xx-xxT19:59:57Z
name: csw.read.only
namespace: default
resourceVersion: "991"
selfLink: /api/v1/namespaces/default/serviceaccounts/e2e.minimal
```

```
uid: ce23da52-a11d-11ea-a990-525400d58002
secrets:
- name: csw.read.only-token-vmvm
```

yaml 出力モードでシークレットをリストすると、トークンが生成されますが、Base64 形式となります（シークレットデータに対する標準の Kubernetes 手順）。Secure Workload は、この形式のトークンを受け入れないため、Base64 からデコードする必要があります。

```
$ kubectl get secret -o yaml csw.read.only-token-vmvmz
apiVersion: v1
data:
ca.crt: ...
namespace: ZGVmYXVsdA==
token: ZXlKaGJHY2lPaUpTVX....HRfZ2JwMVZR
kind: Secret
metadata:
annotations:
kubernetes.io/service-account.name: csw.read.only
kubernetes.io/service-account.uid: ce23da52-a11d-11ea-a990-525400d58002
creationTimestamp: 2020-05-28T19:59:57Z
name: csw.read.only-token-vmvmz
namespace: default
resourceVersion: "990"
selfLink: /api/v1/namespaces/default/secrets/csw.read.only-token-vmvmz
uid: ce24f40c-a11d-11ea-a990-525400d58002
type: kubernetes.io/service-account-token
```

シークレットをリストし、`.data.token` フィールドのみを出力し、1 つのコマンドで Base 64 エンコーディングからデコードするには、`--template` オプションを使用する次のコマンドが役立ちます。

```
$ kubectl get secret csw.read.only-token-vmvmz --template "{{ .data.token }}" |
‹→base64 -d
```

この認証トークンは、Secure Workload UI で Kubernetes オーケストレータを構成するために、ユーザー名/パスワードまたはキー/証明書の代わりに使用できます。

EKS 固有の RBAC 考慮事項

eks_rbac を参照してください。

オーケストレータにより生成されるラベル

「Kubernetes クラスタに関連するラベル」を参照してください。

ホストネットワークモードで実行されている Kubernetes Nginx Ingress コントローラでのポリシーの適用

Cisco Secure Workload は、Kubernetes Ingress オブジェクトを使用して外部クライアントにポッドが公開されるときに、Nginx Ingress コントローラとバックエンドポッドの両方でポリシーを適用します。

（注） Ingress コントローラがホストネットワークモードで実行されていない場合は、IngressControllerAPI を参照してください。

（注） IBM-ICP は、デフォルトで Kubernetes Nginx Ingress コントローラを使用し、ホストネットワークモードのコントロールプレーンノードで実行されます。

Kubernetes Nginx Ingress コントローラを使用してポリシーを適用する手順は次のとおりです。

ステップ 1 こちらの説明に従って、Kubernetes/OpenShift の外部オーケストレータを作成します。

```
→  ~
→  ~ k8s get ingress
NAME           HOSTS   ADDRESS          PORTS   AGE
test-ingress   *       192.168.60.100   80      7s
→  ~
```

ステップ 2 Kubernetes クラスタに入力オブジェクトを作成します。入力オブジェクトの作成に使用される yaml ファイルのスナップショットを次の図に示します。

```
▶ k8s get ingress
NAME                     HOSTS   ADDRESS         PORTS   AGE
svc-ce2e-teeksitlbiwlc   *       192.168.10.13   80      74s
```

```
~
▶ k8s get ingress -o yaml
apiVersion: v1
items:
- apiVersion: extensions/v1beta1
  kind: Ingress
  metadata:
    annotations:
      virtual-server.f5.com/ip: 192.168.10.13
      virtual-server.f5.com/partition: k8scluster
    creationTimestamp: "2020-06-26T21:31:01Z"
    generation: 1
    labels:
      e2e-test: "yes"
    name: svc-ce2e-teeksitlbiwlc
    namespace: default
    resourceVersion: "1074475"
    selfLink: /apis/extensions/v1beta1/namespaces/default/ingresses/svc-ce2e-teeksitlbiwlc
    uid: 5526b4a3-b7f4-11ea-aa09-525400d58002
  spec:
    backend:
      serviceName: svc-ce2e-teeksitlbiwlc
      servicePort: 80
  status:
    loadBalancer:
      ingress:
      - ip: 192.168.10.13
kind: List
metadata:
  resourceVersion: ""
  selfLink: ""
```

ステップ 3 Kubernetes クラスタに Kubernets Nginx Ingress コントローラを展開します。IBM-ICP Ingress コントローラポッドは、デフォルトでコントロールプレーンノードで実行されています。

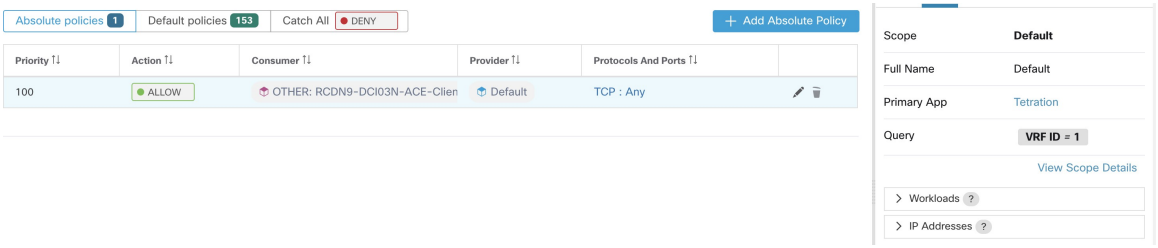
```
~
▶ k8s get pods -o wide -n ingress-nginx
NAME                                       READY   STATUS    RESTARTS   AGE     IP             NODE                          NOMINATED NODE
nginx-ingress-controller-6bc9c6745c-scfzs   1/1     Running   0          2m11s   192.168.10.13   enforcement-scale-16-kube3   <none>

~
▶ k8s get node enforcement-scale-16-kube3 -o wide
NAME                          STATUS   ROLES    AGE    VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE            KERNEL-VERSION      CONTAINER-RUNTIME
enforcement-scale-16-kube3   Ready    <none>   7d5h   v1.12.3   192.168.10.13   <none>        Ubuntu 16.04.5 LTS   4.4.0-139-generic   docker://18.6.1
```

ステップ 4 クラスタ外のコンシューマがアクセスするバックエンドサービスを作成します。次の例では、簡単な *svc-ce2e-teeksitlbiwlc*（http-echo）サービスを作成しています。

```
~
▶ k8s get svc svc-ce2e-teeksitlbiwlc
NAME                     TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
svc-ce2e-teeksitlbiwlc   ClusterIP   10.102.30.231   <none>        80/TCP    6m11s
```

ステップ 5 外部コンシューマとバックエンドサービスの間にポリシーを作成します。[ポリシーの適用（Policy Enforcement）] タブを使用してポリシーを適用します。



ステップ 6 Nginx Ingress コントローラの場合、Secure Workload ソフトウェアでは、送信元は上記の手順で指定されたコンシューマであり、宛先は対応する Ingress コントローラポッド IP である、適切な許可/ドロップルールが適用されます。バックエンドポッドの場合、Secure Workload ソフトウェアでは、送信元は入力ポッドであり、宛先はバックエンドポッド IP である、適切な許可/ドロップルールが適用されます。

Deployment/Daemonset として実行されている Kubernetes Nginx/Haproxy Ingress コントローラでのポリシーの適用

Cisco Secure Workload は、Kubernetes Ingress オブジェクトを使用して外部クライアントにポッドが公開されるときに、Ingress コントローラとバックエンドポッドの両方でポリシーを適用します。

Ingress コントローラにポリシーを適用する手順は次のとおりです。

ステップ 1 OpenAPI を使用して、Kubernetes/OpenShift の外部オーケストレータを作成/更新します。OpenAPI を使用して外部オーケストレータを作成する方法については、「オーケストレータ」を参照してください。外部オーケストレータ設定のための Ingress コントローラの情報を追加します。

ステップ 2 Kubernetes クラスタで Ingress オブジェクトを作成します。

ステップ 3 Kubernetes クラスタで Ingress コントローラを展開します。

ステップ 4 クラスタ外のコンシューマがアクセスするバックエンドサービスを作成します。

ステップ 5 外部コンシューマとバックエンドサービスの間にポリシーを作成します。[ポリシー適用（Policy Enforcement）] タブを使用してポリシーを適用します。

ステップ 6 Ingress コントローラの場合、Secure Workload ソフトウェアでは、送信元は上記の手順で指定されたコンシューマであり、宛先は対応する Ingress コントローラポッド IP である、適切な許可/ドロップルールが適用されます。バックエンドポッドの場合、Secure Workload ソフトウェアでは、送信元は入力ポッドであり、宛先はバックエンドポッド IP である、適切な許可/ドロップルールが適用されます。

トラブルシューティング

- クライアントキー/証明書の資格情報の解析/不一致

これらは PEM 形式で提供し、kubectl.conf ファイルからの正しいエントリである必要があります。お客様が CA 証明書をクライアント証明書フィールドに貼り付けたり、キーと証明書が互いに一致していないことがありました。

- GKE 認証情報の代わりに gcloud 認証情報

  gcloud CLI で GKE を使用しているお客様が、GKE クラスタの認証情報が必要な場合に、誤って gcloud 認証情報を提供します。

- Kubernetes クラスタのバージョンがサポートされていません

  互換性のないバージョンの Kubernetes を使用すると、エラーが発生する可能性があります。Kubernetes のバージョンが、サポートされているバージョンのリストにあることを確認してください。

- 資格情報に十分な権限がありません

  使用されている認証トークンや、ユーザーまたはクライアントのキー/証明書に、上記の表にリストされているすべての権限があることを確認します。

- Kubernetes インベントリが変動し続けています

  hosts_list フィールドは、同じ Kubernetes クラスタの API サーバーのプールを指定します。これを使用して複数の Kubernetes クラスタを設定することはできません。Secure Workload は稼働状態を調べ、これらのエンドポイントの 1 つをランダムに選択して接続し、Kubernetes インベントリ情報を取得します。ここではロードバランシングは実行されず、これらのエンドポイント間で負荷が均等に分散される保証もありません。これらが異なるクラスタである場合、どのクラスタの API サーバーに接続するかに応じて、Kubernetes インベントリはそれらのクラスタ間を切り替え続けます。

- 複数の認証方式

  複数の認証方式（ユーザー名/パスワード、認証トークン、クライアントキー/証明書）が設定中に入力される場合があり、API サーバーとの間で確立されたクライアント接続で使用されます。ここでは、有効な同時認証方式に対する標準の Kubernetes ルールが適用されます。

- SSL 証明書の検証失敗

  Kubernetes API エンドポイントが NAT またはロードバランサの背後にある場合、kube コントロールプレーンノードで生成された SSL 証明書の DN が、Cisco Secure Workload で設定された IP アドレスと一致しない場合があります。これにより、CA 証明書が提供され、有効であっても、SSL 検証が失敗します。Insecure ノブは厳密なサーバー SSL 証明書の検証をバイパスするため、この問題を回避するのに役立ちますが、MITM の問題につながる可能性があります。これに対する正しい修正は、CA 証明書を変更して、Kubernetes クラスタへの接続に使用できるすべての DNS/IP エントリに SAN（サブジェクト代替名）エントリを指定することです。

VMware vCenter

vCenter 統合により、ユーザーは設定された vCenter からベアメタルおよび VM 属性を取得できます。

「vCenter」タイプに外部オーケストレータ設定が追加されると、Secure Workload はその vCenter インスタンスによって制御されるすべてのベアメタルと VM について、ベアメタル属性と VM 属性を取得します。Secure Workload はベアメタル/VM の次の属性をインポートします。- a) ホスト名 b) IP アドレス c) BIOS UUID d) カテゴリ/ラベル。

インベントリがアプライアンスに存在しない場合、新しいインベントリが上記のベアメタル/VM 属性を使用して Secure Workload で作成されます。インベントリがアプライアンスに既に存在する場合（ベアメタル/VM で実行されている Secure Workload 可視性センサーによって作成）、既存のインベントリは、取得されたベアメタル/VM のカテゴリ/ラベルリストでラベル付けされます。

前提条件

- 接続に必要な場合、Secure Connector トンネル。
- サポートされている vCenter バージョンが 6.5 以降であること

設定フィールド

「外部オーケストレータの作成」で説明されている一般的な設定フィールドのほかに、次のフィールドを設定できます。

- [ホストリスト（Hosts List）] は、ベアメタル/VM 属性がフェッチされる vCenter サーバーを指すホスト名/IP とポートのペアの配列です。

ワークフロー

- ユーザーは Secure Workload クラスタから該当 IP やポートを使用して vCenter サーバーにアクセスできることを最初に確認する必要があります。
- TaaS の場合、または vCenter サーバーに直接アクセスできない場合、ユーザーはセキュアなコネクタトンネルを確立して接続を提供する必要があります。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、vCenter サーバーから学習したすべての VM に次のラベルを追加します。

キー	値
orchestrator_system/orch_type	vCenter
orchestrator_system/cluster_name	*<Name given to this cluster's configuration>*
orchestrator_system/cluster_id	*<UUID of the cluster's configuration in |product|>*

インスタンス固有のラベル

次のラベルはインスタンス固有です。

表 *2:* 次のラベルはインスタンス固有です。

キー	値
orchestrator_system/workload_type	vm
orchestrator_system/machine_id	ベアメタル/VM の BIOS UUID
orchestrator_system/machine_name	ベアメタル/VM のホスト名
orchestrator_'<Category Name>'	<Tag Value>

警告

- vCenter 用の外部オーケストレータ構成が追加されると、Secure Workload ソフトウェアはホストリストで指定された vCenter サーバーに接続します。サーバーへの接続が成功すると、Secure Workload ソフトウェアは、vCenter サーバーに存在するすべてのベアメタルと仮想マシンのホスト名、IP アドレス、およびカテゴリ/ラベルをインポートします。ベアメタルと VM のホスト名と IP アドレスをインポートするには、すべてのベアメタルと VM に VM ツールをインストールする必要があります。特定のベアメタル/仮想マシンに VM ツールがインストールされていない場合、Secure Workload ソフトウェアはその特定のベアメタル/VM のカテゴリ/ラベルを表示しません。
- Cisco Secure Workload ソフトウェアは、ベアメタル/VM のカスタム属性をインポートしません。
- vCenter サーバーの負荷を軽減するために、[デルタ（Delta）] インターバルタイマーを 10 分以上に設定することをお勧めします。上記のタイマーが変更されると、vCenter サーバー上のインベントリ/ラベルに変更がある場合に、少なくとも 10 分の伝播遅延が発生します。

トラブルシューティング

- 接続の問題

Secure Workload アプライアンスが vCenter サーバーに接続または到達できない場合、外部オーケストレータの [接続ステータス（Connection Status）] タブに、適切なエラー（エラーがある場合）とともに障害ステータスが表示されます。

- Cisco Secure Workload ソフトウェアのヘルスチェック。

  [メンテナンス/サービスステータス（MAINTENANCE/Service Status）] ページをチェックして、サービスが停止していないか確認してください。OrchestratorInventoryManager が稼働しているか確認してください。

DNS

DNS 統合により、Secure Workload は、CNAME および A/AAAA レコードからのホスト名などの DNS 情報で、既知のインベントリに注釈を付けることができます。

「dns」タイプの外部オーケストレータ構成が追加されると、Secure Workload アプライアンスは DNS サーバーへの接続を試み、DNS レコードのゾーン転送ダウンロードを実行します。これらのレコード（A/AAAA および CNAME レコードのみ）は解析され、「orchestrator_system/dns_name」と呼ばれる単一の複数値ラベルを使用して、（オーケストレータが設定されているテナントに属する）Secure Workload パイプラインのインベントリを強化するために使用されます。値は、その IP アドレスを（直接的または間接的に）指す DNS エントリになります。

前提条件

- セキュアコネクタトンネル（接続に必要な場合）
- サポートされている DNS サーバー：BIND9、AXFR（RFC 5936）をサポートするサーバー、Microsoft Windows Server 2016

設定フィールド

- [DNSゾーン（DNS zones）] は文字列配列であり、各文字列は DNS サーバーから転送される DNS ゾーンを表します。すべての DNS ゾーンには、末尾にピリオド（「.」）の文字が必要です。
- [ホストリスト（Hosts List）] はホスト名/IP とポートのペアの配列であり、DNS レコードを取得する DNS サーバーを指します。ここでは、HA のみを目的として複数の DNS サーバーを構成できます。hosts_list で複数の DNS サーバーが指定された場合、高可用性では「最初の正常なサーバー」が使用されます。hosts_list のエントリの先頭から順に優先されます。ゾーンを DNS サーバー間で分割することはできません。

ワークフロー

- まず、ユーザーは、Secure Workload クラスタから該当 IP/ポートで DNS サーバーに到達可能なことを確認する必要があります。
- TaaS の場合、または DNS サーバーに直接到達できない場合、ユーザーは安全なコネクタトンネルを設定して接続を提供する必要があります。
- DNS サーバーで正しい DNS ゾーン転送 ACL または構成を設定します。詳細については、個別の DNS サーバーソフトウェアのマニュアルを参照してください。

生成されたラベル

orchestrator_system/dns_name -> その値がすべてその IP を示す CNAME および A/AAAA ホスト名である複数の値フィールド。

警告

- DNS オーケストレータフィードはメタデータフィードです。DNS ゾーン転送から学習された IP アドレスによって Cisco Secure Workload にインベントリアイテムが作成されることはありません。代わりに、既存の IP アドレスのラベルが新しい DNS メタデータで更新されます。不明な IP の DNS データは、警告なしで破棄されます。センサー統合または他のオーケストレータ統合を介して学習されていない IP の DNS メタデータに注釈を付けるには、CMDB 一括アップロードメカニズムを使用して IP をアップロードし、IP のインベントリエントリを作成する必要があります。CMDB アップロードから学習されたサブネットによって、インベントリエントリが作成されることはありません。
- DNS サーバーからの CNAME レコードと A/AAAA レコードのみが処理されます。CNAME レコードは、それらが指す A/AAAA レコードを介して最終的な IPv4/IPv6 レコードに処理されます。CNAME が同じオーケストレータからの A/AAAA レコードを指している限り、単一レベルの参照のみがサポートされます（つまり、CNAME -> CNAME -> A/AAAA やそれ以上のチェーンは参照されません）。異なる DNS オーケストレータ間での CNAME 参照はサポートされていません。

トラブルシューティング

- 接続の問題

  Cisco Secure Workload は、Secure Workload アプライアンスサーバーの 1 つから、または TaaS の場合はクラウドから、または Secure Workload Secure Connector VPN トンネルサービスをホストしている VM から発信される TCP 接続を使用して、指定された IP/ホスト名とポート番号への接続を試みます。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。

- DNS AXFR 特権の問題

さらに、ほとんどの DNS サーバー（BIND9 または Windows DNS または Infoblox）では、クライアント IP が DNS ゾーン転送（DNS プロトコルのオペコードによる AXFR 要求）を試みるときに、追加の設定が必要になります。これは、個々の DNS レコードを解決するためのシンプルな DNS 要求と比べて、必要なリソース消費量と特権がより多くなるためです。これらのエラーは、通常、AXFR が理由コード 5（REFUSE）で拒否されたときに表示されます。

したがって、DNS サーバーが正しく設定されていることを確認するための手動テストは、ホスト名ルックアップが成功したかによるのではなく、（dig などのツールを使用して）具体的な AXFR 要求をテストする必要があります。

DNS サーバーからの AXFR ゾーン転送の実行に失敗すると、Secure Workload アプライアンスによって「authentication_failure_error」フィールドで報告されます。

また、Secure Workload は、設定されたすべての DNS ゾーンからのゾーン転送を試みますが、DNS データを Secure Workload ラベルデータベースに挿入するためには、そのすべてが成功する必要があることに注意してください。

- インベントリの [ホスト名（Hostname）] フィールドは [DNS] フィールドによって入力されず、「hostname」は常に Secure Workload センサーから学習されます。インベントリがセンサーからではなく、CMDB アップロードを介してアップロードされた場合、ホスト名が欠落している場合があります。DNS オーケストレータ ワークフローからのすべてのデータは、「orchestrator_system/dns_name」ラベルの下にのみ表示され、ホスト名フィールドに入力されることはありません。

DNS オーケストレータのフル/差分ポーリングの動作

デフォルトのフルスナップショット間隔は 24 時間です

デフォルトの差分スナップショット間隔は 60 分です

デフォルトの間隔は、各タイマーの最小許容値でもあります。

DNS レコードはほとんど変更されないため、最適なフェッチ動作のために、Secure Workload は差分スナップショット間隔ごとに、いずれかの DNS ゾーンのシリアル番号が前の間隔から変更されているかチェックします。ゾーンが変更されていない場合、アクションは必要ありません。

いずれかのゾーンが変更された場合は、構成されているすべての DNS ゾーン（変更された単一のゾーンだけでなく）からゾーン転送を実行します。

Secure Workload は、完全なスナップショット間隔ごとに、ゾーンのシリアル番号が変更されたかどうかに関係なく、すべてのゾーンからゾーン転送ダウンロードを実行して、ラベルデータベースに挿入します。

サポートされない機能

警告

- DNAME エイリアスとルックアップはサポートされていません。
- 増分ゾーン転送 (IXFR) はサポートされていません。

Infoblox

Infoblox 統合により、Secure Workload は Infoblox サブネット、ホスト（record:host）、および A/AAAA レコードを Secure Workload インベントリデータベースにインポートできるようになります。拡張可能な属性の名前と値はそのままインポートされ、範囲と適用ポリシーを定義するための Secure Workload ラベルとして使用できます。

（注） 拡張可能な属性を持つ Infoblox オブジェクトのみが考慮されます。すなわち、拡張可能な属性がアタッチされていないものは、インポートから除外されます。

下の図は、Infoblox からインポートされた、拡張可能な属性 Department を持つホストオブジェクトに対して生成されたラベルの例を示しています。

図 *12 : Infoblox* ラベルの例

```
1. orchestrator_Department = AES789
2. orchestrator_system/cluster_id = [redacted]
3. orchestrator_system/cluster_name = scale13-ib
4. orchestrator_system/machine_id =
   record:host/[redacted]:client8/%20
5. orchestrator_system/machine_name = client8
6. orchestrator_system/orch_type = infoblox
```

前提条件

- WAPI バージョン 2.6、2.6.1、2.7、2.7.1 をサポートする Infoblox REST API エンドポイント（推奨）

設定フィールド

「外部オーケストレータの作成」で説明されている共通の設定フィールドのほかに、次のフィールドを設定できます。

共通フィールド	必須	説明
[ホストリスト（Hosts List）]	○	ホストリストは、1 つの Infoblox グリッドを示します。つまり、REST API アクセス権を持つ複数のグリッドメンバーを追加できます。接続エラーが発生した場合、外部オーケストレータはリスト内の次のメンバーに切り替えます。別の Infoblox グリッドからラベルをインポートする場合は、新しい外部オーケストレータを作成してください。

（注） Infoblox 外部オーケストレータの場合、IPv4 および IPv6（デュアルスタックモード）アドレスがサポートされます。ただし、デュアルスタックのサポートはベータ版の機能であることに注意してください。

ワークフロー

- まず、ユーザーは、Secure Workload クラスタから Infoblox REST API エンドポイントに到達できることを確認する必要があります。
- TaaS の場合、または Infoblox サーバーに直接到達できない場合、ユーザーは Secure Connector トンネルを設定して接続を可能にする必要があります。
- *Infoblox* タイプの外部オーケストレータを作成します。Infoblox データの量（サブネット、ホスト、および A/AAAA レコードの数）によっては、最初の完全なスナップショットが Cisco Secure Workload で使用可能になるまでに最大 1 時間かかります。
- Infoblox 設定の際に、ユーザーはいずれかのレコードタイプ（サブネット、ホスト、A/AAAA レコード）の選択を解除することもできます。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、Infoblox から取得したすべてのオブジェクトに次のシステムラベルを追加します。

キー	値
orchestrator_system/orch_type	InfoBlox

キー	値
orchestrator_system/cluster_id	<UUID of the external orchestrator in Secure Workload
orchestrator_system/cluster_name	<Name given to this external orchestrator>
orchestrator_system/machine_id	<Infoblox object reference/identifier>
Orchestrator_system/machine_name	<Infoblox host (DNS) name>

生成されるラベル

すべての Infoblox 拡張可能属性は、プレフィックス *orchestrator_* の付いた Secure Workload ラベルとしてインポートされます。たとえば、*Department* という拡張可能属性を持つホストは、Secure Workload のインベントリ検索で *orchestrator_Department* としてアドレス指定できます。

キー	値
orchestrator_<extensible attribute>	<value(s) of the extensible attribute as retrieved from Infoblox>

警告

- Infoblox からインポートできるサブネットの最大数は 50000 です。
- Infoblox からインポートできるホストと A/AAAA レコードの最大数は、合計で 400000 です。

トラブルシューティング

- 接続の問題：Secure Workload は、Secure Workload アプライアンスサーバーの 1 つから、または TaaS の場合はクラウドから、または Secure Workload セキュア コネクタ トンネル サービスをホストしている VM から、HTTPS 接続を使用して、指定された IP/ホスト名とポート番号に接続しようとします。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。また、指定されたクレデンシャルが正しいこと、および REST API 要求を Infoblox に送信するための権限があることを確認してください。
- すべての予期されるオブジェクトがインポートされない：Secure Workload ではサブネット、ホスト、および拡張可能な属性が添付された A/AAAA レコードのみがインポートされます。Infoblox からインポートできるオブジェクトの数には制限があることに注意してください（「注意事項」を参照）。

- インベントリでサブネットが見つからない：Secure Workload インベントリは設計上 IP アドレスのみが含まれているため（ホストや A/AAAA レコードなど）、インベントリ検索を使用して Infoblox サブネットを見つけることはできません。
- ホストまたは A/AAAA レコードが見つからない：Secure Workload は Infoblox から取得したすべての拡張可能属性をインポートします。たとえば、インベントリ検索でプレフィックス *orchestrator_* を拡張可能な属性名に追加してください。サブネットの拡張可能属性は、Infoblox で継承とマークされていない場合、ホストの一部ではないため、Cisco Secure Workload で検索できないことに注意してください。

F5 BIG-IP

F5 BIG-IP 統合により、Secure Workload は、F5 BIG-IP ロード バランサ アプライアンスから仮想サーバーをインポートし、サービスインベントリを取得できます。サービスインベントリは F5 BIG-IP 仮想サーバーに対応し、そのサービスは VIP（仮想 IP アドレス）、プロトコル、およびポートによって特性が決まります。Secure Workload にインポートされると、このサービスインベントリには service_name などのラベルが付けられます。これは、インベントリ検索で使用できるだけでなく、Secure Workload の範囲とポリシーを作成するためにも使用できます。

この機能の大きな利点は、F5 BIG-IP の外部オーケストレータが Secure Workload ポリシーを仮想サーバーに割り当てられたセキュリティルールに変換し、REST API を介して F5 BIG-IP ロードバランサに展開するという仕方でポリシーを適用できることです。

前提条件

- セキュアコネクタトンネル（接続に必要な場合）
- F5 BIG-IP REST API エンドポイントバージョン 12.1.1

設定フィールド

「外部オーケストレータの作成」で説明されている一般的な構成フィールドのほかに、次のフィールドを構成できます。

フィールド	必須	説明
ホストのリスト	対応	このフィールドでは、F5 BIG-IP ロードバランサの REST API エンドポイントを指定します。F5 BIG-IP 用に高可用性が設定されている場合は、他のメンバーノードも入力してください。現在のノードとの通信に失敗した場合、外部オーケストレータによって切り替えられます。別の F5 BIG-IP ロードバランサからラベルをインポートする場合は、新しい外部オーケストレータを作成する必要があります。
適用の有効化（Enable Enforcement）	×	デフォルト値は false（チェックなし）です。チェックすると、Secure Workload のポリシーの適用が許可され、対応する F5 BIG-IP ロードバランサにセキュリティポリシールールを展開できます。指定されたログイン情報には、F5 BIG-IP REST API の書き込みアクセス権が必要であることに注意してください。
ルートドメイン	×	デフォルト値は 0（ゼロ）です。ルートドメインは、外部オーケストレータによって考慮される仮想サーバーを指定します。仮想サーバーは、特定のルートドメインに割り当てられたパーティションのリストによって決定され、それらのパーティションで定義された仮想サーバーのみが Cisco Secure Workload にインポートされます。

ワークフロー

- まず、ユーザーは、Cisco Secure Workload から F5 BIG-IP REST API エンドポイントに到達できることを確認する必要があります。
- TaaS の場合、または F5 BIG-IP アプライアンスに直接到達できない場合、ユーザーは Secure Connector トンネルを設定して接続を可能にする必要があります。
- *F5 BIG-IP* タイプの外部オーケストレータを作成します。
- 差分間隔の値によっては、F5 BIG-IP 仮想サーバーの最初の完全スナップショットが完了するまでに最大 60 秒（デフォルトの差分間隔）かかる場合があります。その後、生成されたラベルを使用して、Secure Workload の範囲と適用ポリシーを作成できます。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、*F5 BIG-IP* の外部オーケストレータに次のシステムラベルを追加します。

キー	値
orchestrator_system/orch_type	f5
orchestrator_system/cluster_id	<外部オーケストレータの UUID>
orchestrator_system/cluster_name	<外部オーケストレータに割り当てられた名前>
orchestrator_system/workload_type	サービス
orchestrator_system/namespace	<仮想サーバーが属するパーティション>
orchestrator_system/service_name	<F5 BIG-IP 仮想サーバー名>

生成されるラベル

外部オーケストレータは、仮想サーバーごとに次のラベルを生成します。

キー	値
orchestrator_annotation/snat_address	<仮想サーバーの SNAT アドレス>

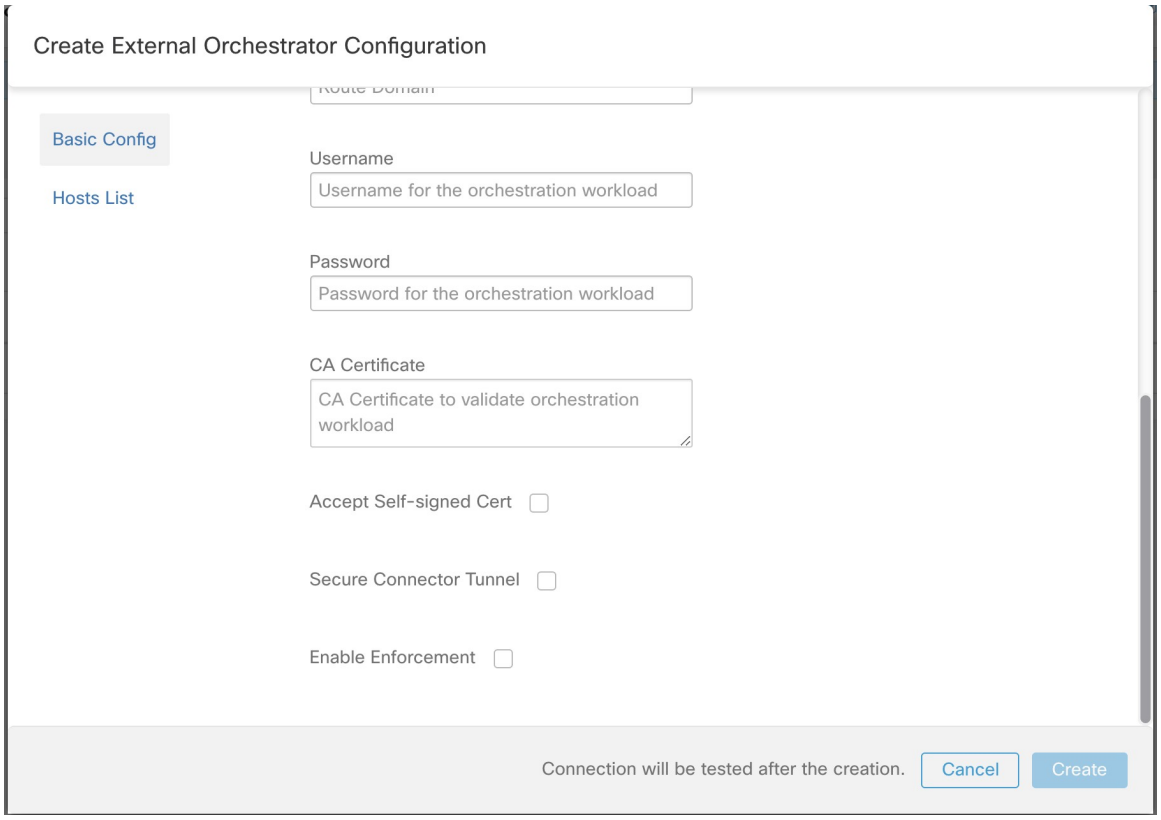
ポリシーの実施

この機能により、Secure Workload は、ラベル付きの F5 BIG-IP 仮想サーバーに一致するプロバイダーグループの論理ポリシーを F5 BIG-IP セキュリティポリシールールに変換し、それらのルールを REST API を使用してロード バランサ アプライアンスに展開できます。前述のよう

に、それぞれの F5-BIGP 仮想サーバーへの既存のセキュリティポリシーの割り当ては、Secure Workload によって生成されたセキュリティポリシーを指す新しい割り当てに置き換えられます。ユーザーが作成したすべてのセキュリティポリシーは、F5-BIGIP ポリシーリストから操作または削除されません。

デフォルトでは、以下の図に示すように、[オーケストレータの作成（Create Orchestrator）] ダイアログで、[適用の有効化（Enable Enforcement）] フィールドはオンになっておらず、適用は無効化されています。

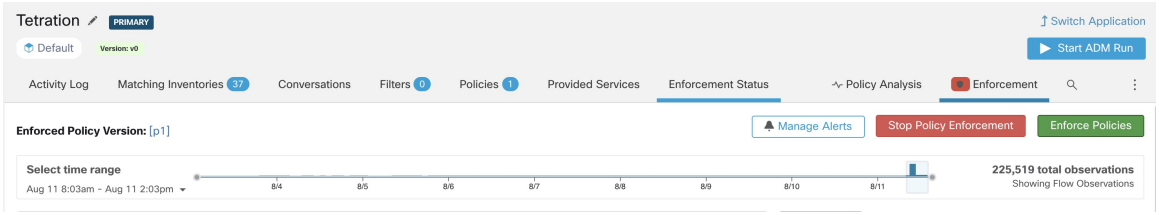
図 *13* : 設定オプション *[*適用の有効化（*Enable Enforcement*）*]*



指定されたチェックボックスをクリックするだけで、オーケストレータの適用を有効にすることができます。このオプションは、必要に応じていつでも変更できます。

オーケストレータ設定の作成または編集によって有効にされたかどうかに関係なく、オーケストレータの適用を有効にしても、現在の論理ポリシーがロード バランサ アプライアンスにすぐに展開されることはありません。このタスクは、次の図に示すように、ユーザーによって、またはインベントリの更新によってトリガーされるワークスペースポリシー適用の一環として実行されます。ただし、オーケストレータの適用を無効にすると、展開されたすべてのセキュリティポリシールールが F5-BIGP ロードバランサからすぐに削除されます。

図 *14* : ワークスペースポリシーの適用



（注）

- F5 BIG-IP のオーケストレータも、セキュリティポリシールールの逸脱を検出し、Secure Workload ポリシーに置き換えます。そのため、仮想サーバーに対するポリシーの変更は、すべて Secure Workload を使用して行う必要があります。
- ポリシーの適用が停止されるか、外部オーケストレータが削除されると、すべての Secure Workload ポリシーが F5 BIG-IP ロードバランサから削除されるため、仮想サーバーのセキュリティポリシーは空になります。

外部オーケストレータの OpenAPI ポリシー適用ステータスを使用して、外部オーケストレータに関連付けられたロード バランサ アプライアンスへの Secure Workload ポリシー適用のステータスを取得できます。この機能は、F5-BIGIP アプライアンスへのセキュリティポリシールールの展開が成功したかどうかを確認するのに役立ちます。

F5 入力コントローラのポリシーの適用

Cisco Secure Workload は、ポッドが Kubernetes 入力オブジェクトを使用して外部クライアントに公開されるときに、*F5 BIG-IP* ロードバランサとバックエンドポッドの両方でポリシーを適用します。

F5 入力コントローラを使用したポリシー適用手順は次のとおりです。

ステップ 1 前述したように、*F5 BIG-IP* ロードバランサの外部オーケストレータを作成します。

ステップ 2 ここに記載する説明に従って、Kubernetes または OpenShift の外部オーケストレータを作成します。

```
→  ~
→  ~ k8s get ingress
NAME           HOSTS   ADDRESS          PORTS   AGE
test-ingress   *       192.168.60.100   80      7s
→  ~
```

ステップ 3 Kubernetes クラスタに入力オブジェクトを作成します。入力オブジェクトの作成で使用する yaml ファイルのスナップショットを次の図に示します。

```
→  ~
→  ~ k8s get ingress test-ingress -o yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    virtual-server.f5.com/ip: 192.168.60.100
    virtual-server.f5.com/partition: k8scluster
  creationTimestamp: "2019-07-26T18:34:39Z"
  generation: 1
  name: test-ingress
  namespace: default
  resourceVersion: "8310"
  selfLink: /apis/extensions/v1beta1/namespaces/default/ingresses/test-ingress
  uid: 06f8a705-afd4-11e9-97fb-525400d58002
spec:
  backend:
    serviceName: nginx
    servicePort: 80
status:
  loadBalancer:
    ingress:
    - ip: 192.168.60.100
→  ~
```

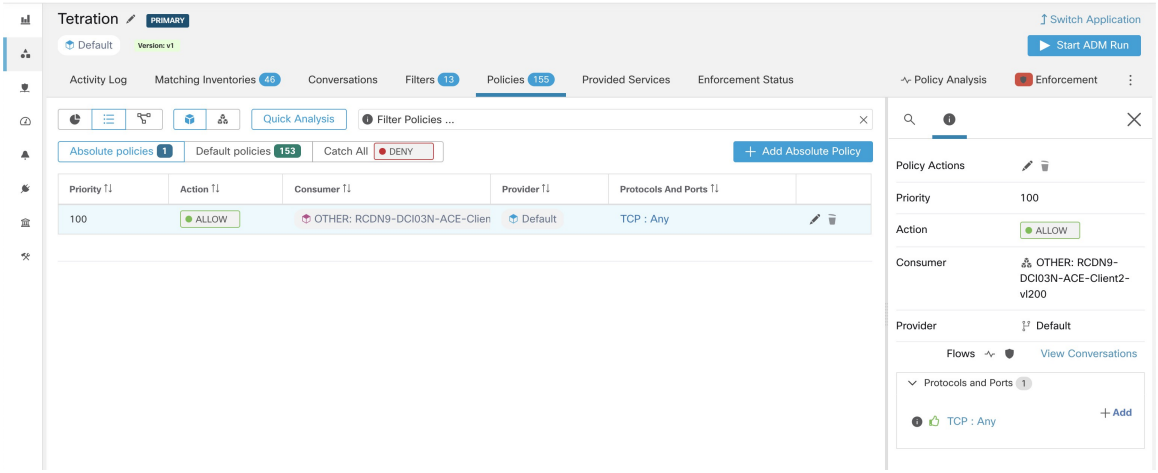
ステップ 4 F5 入力コントローラポッドを Kubernetes クラスタに展開します。

```
→  ~ k8s get deploy -n kube-system
NAME                     DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
coredns                  2         2         2            2           31m
k8s-bigip-ctlr-cluster   1         1         1            1           5m20s
→  ~
```

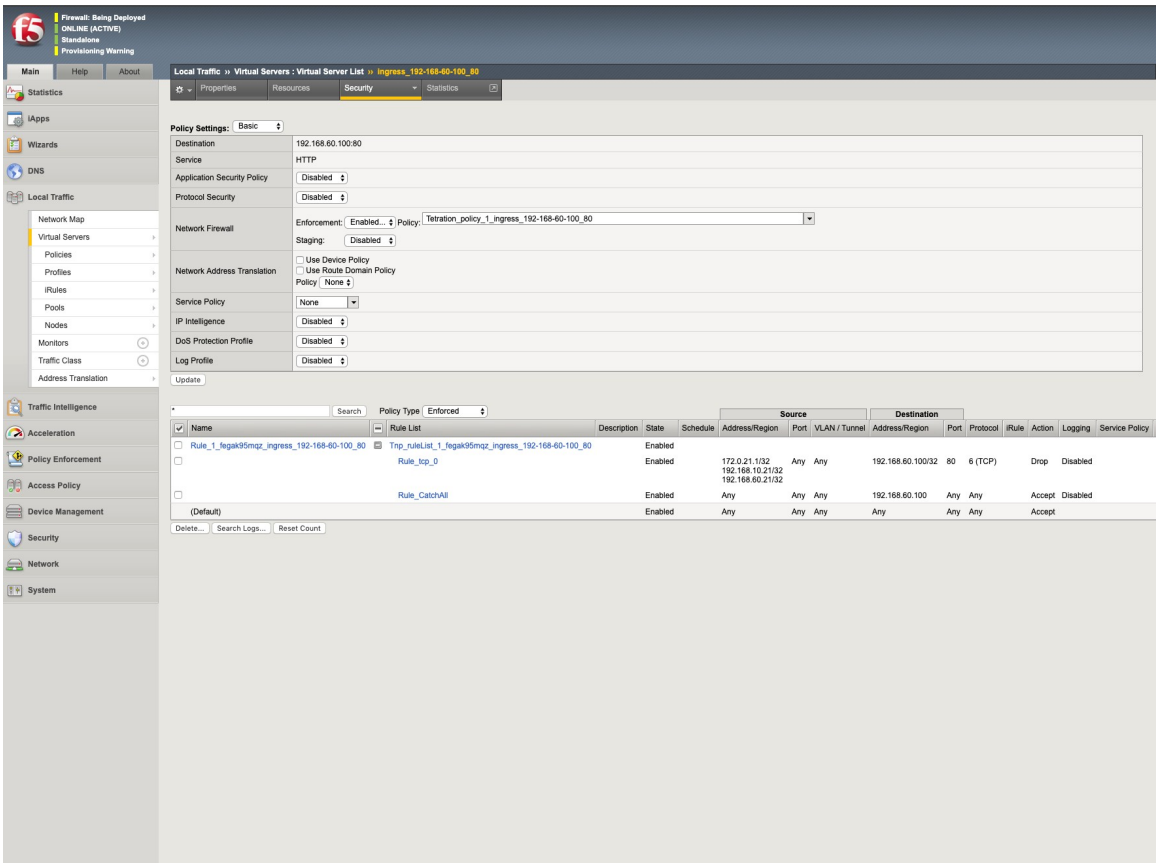
ステップ 5 クラスタ外のコンシューマがアクセスするバックエンドサービスを作成します。以下の例では、*nginx* サービスを作成しています。

```
→  ~
→  ~ k8s get deploy
NAME    DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx   1         1         1            0           5s
→  ~
```

ステップ 6 外部コンシューマとバックエンドサービス間にポリシーを作成します。[ポリシー適用（Policy Enforcement）] タブを使用してポリシーを適用します。



ステップ 7 *F5 BIG-IP* ロードバランサとバックエンドポッドのポリシーを確認します。F5 ロードバランサの場合、Secure Workload は適切な許可またはドロップルールを適用します。送信元はステップ 6 で指定したコンシューマであり、宛先は VIP（F5 の入力仮想サービスの VIP）になります。バックエンドポッドの場合、Secure Workload は適切な許可またはドロップルールを適用します。このとき、送信元が SNIP（SNAT プールが有効になっている場合）またはF5IP（自動マップが有効になっている場合）になり、宛先がバックエンドポッド IP になります。



警告

- F5 BIG-IP HA モードの展開フェーズ中に、構成同期オプションを有効にしてください。有効にすると、外部オーケストレータが、現在接続しているホストから仮想サーバーの最新リストをフェッチできます。
- F5 BIG-IP HA 展開 モードの場合、SNAT プールの代わりに Auto-Map がアドレス変換用に構成されている場合は、プライマリ BIG-IP がフローティングセルフ IP アドレスで構成されていることを確認してください。
- 単一のアドレスとして指定された VIP のみがサポートされています。つまり、サブネットとして指定された VIP はサポートされていません。

トラブルシューティング

- 接続の問題 Secure Workload は、Secure Workload アプライアンスサーバーの 1 つから、または TaaS の場合はクラウドから、または Secure Workload セキュア コネクタ トンネル サービスをホストしている VM から、HTTPS 接続を使用して、指定された IP/ホスト名とポート番号に接続しようとします。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。また、指定されたクレデンシャルが正しいこと、および F5 BIG-IP アプライアンスに REST API 要求を送信するための読み取りおよび書き込みアクセス権限があることを確認してください。
- セキュリティルールが見つからない定義された仮想サーバーのセキュリティルールが見つからない場合は、ポリシー適用の実行後に、対応する仮想サーバーが有効になっていることを確認してください。その可用性/ステータスが使用可能/有効になっている必要があります。

Citrix Netscaler

Citrix Netscaler 統合により、Secure Workload は、Netscaler ロード バランサ アプライアンスからロードバランシング仮想サーバーをインポートし、サービスインベントリを取得できます。サービスインベントリは、仮想サーバーによって提供される Netscaler サービスに対応し、*service_name* などのラベルを指定されます。これらのラベルは、インベントリ検索で使用することが可能で、Secure Workload の範囲とポリシーの作成に使用できます。

この機能の大きな利点は、*Citrix Netscaler* の外部オーケストレータが Secure Workload ポリシーを Netscaler ACL ルールに変換し、REST API を介して Netscaler ロードバランサに展開するという仕方でポリシーを適用できることです。

前提条件

- セキュアコネクタトンネル（接続に必要な場合）
- Netscaler REST API エンドポイントバージョン 12.0.57.19

設定フィールド

「外部オーケストレータの作成」で説明されている共通の設定フィールドのほかに、次のフィールドを設定できます。

共通フィールド	必須	説明
[ホストリスト（Hosts List）]	○	これは、Citrix Netscaler ロードバランサの REST API エンドポイントを指します。高可用性構成の場合は、他のメンバーノードも入力してください。外部オーケストレータは、現行ノードとの通信に失敗した場合に切り替えます。別の Citrix Netscaler ロードバランサからラベルをインポートする場合は、新しい外部オーケストレータを作成する必要があります。
[適用の有効化（Enable Enforcement）]	×	デフォルト値は false（チェックボックスがオフ）です。チェックボックスをオンにすると、Secure Workload のポリシー適用によって、対応する Citrix Netscaler ロードバランサに ACL ルールを展開できます。 指定した資格情報には、Citrix Netscaler REST API への書き込みアクセス権が必要であることに注意してください。

ワークフロー

- まず、ユーザーは、Secure Workload クラスタから Netscaler REST API エンドポイントに到達できることを確認する必要があります。
- TaaS の場合、または Netscaler アプライアンスに直接到達できない場合、ユーザーは Secure Connector トンネルを設定して接続を提供する必要があります。
- タイプが Citrix Netscaler の外部オーケストレータを作成します。

- デルタ間隔の値によっては、Netscaler仮想サーバーの最初の完全スナップショットが完了するまでに最大 60 秒（デフォルトのデルタ間隔）かかる場合があります。その後、生成されたラベルを使用して、Secure Workload の範囲と適用ポリシーを作成できます。
- Secure Workload からポリシーを適用して、Netscaler ACL ルールを展開します。

オーケストレータにより生成されるラベル

Cisco Secure Workload は、Citrix NetScaler の外部オーケストレータに次のシステムラベルを追加します。

キー	値
orchestrator_system/orch_type	nsbalancer
orchestrator_system/cluster_id	<UUID of the external orchestrator>
orchestrator_system/cluster_name	<Name given to this external orchestrator>
orchestrator_system/workload_type	service
orchestrator_system/service_name	<Name of the load balancing virtual server>

生成されたラベル

負荷分散仮想サーバーごとに、外部オーケストレータは次のラベルを生成します。

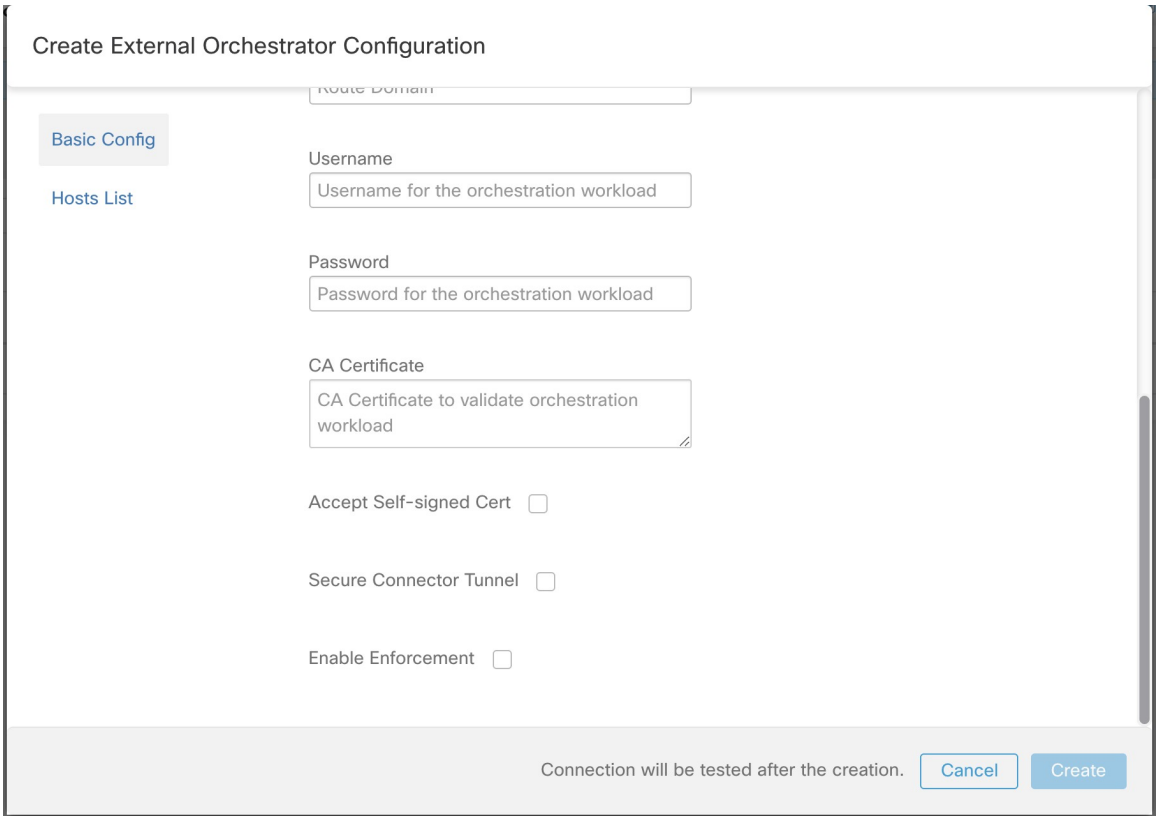
キー	値
orchestrator_annotation/snat_address	<仮想サーバーの SNAT アドレス>

ポリシーの実施

この機能により、Secure Workload は、ラベル付きの *Citrix Netscaler* 仮想サーバーに一致するプロバイダーグループの論理ポリシーを *Citrix Netscaler* ACL ルールに変換し、それらのルールを REST API を使用してロードバランサ アプライアンスに展開できます。前述のように、すべての既存の ACL ルールは、Secure Workload によって生成されたポリシールールに置き換えられます。

デフォルトでは、以下の図に示すように、[オーケストレータの作成（Create Orchestrator）] ダイアログで、[適用の有効化（Enable Enforcement）] フィールドはオンになっておらず、適用は無効化されています。

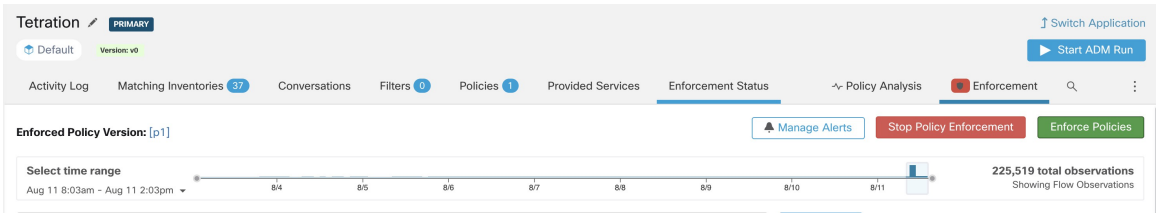
図 *15* : 設定オプション *[*適用の有効化（*Enable Enforcement***）*]***



指定されたチェックボックスをクリックするだけで、オーケストレータの適用を有効にすることができます。このオプションは、必要に応じていつでも変更できます。

オーケストレータ設定の作成または編集によって有効にされたかどうかに関係なく、オーケストレータの適用を有効にしても、現在の論理ポリシーがロード バランサ アプライアンスにすぐに展開されることはありません。このタスクは、次の図に示すように、ユーザーによって、またはインベントリの更新によってトリガーされるワークスペースポリシー適用の一環として実行されます。ただし、オーケストレータの適用を無効にすると、展開されたすべての ACL ルールが *Citrix Netscaler* ロードバランサからすぐに削除されます。

図 *16* : ワークスペースポリシーの適用



（注）

- *Citrix Netscaler* のオーケストレータも、ACL ルールの逸脱を検知し、Secure Workload ポリシーに置き換えます。そのため、ロードバランシング仮想サーバーに対するポリシーの変更は、すべて Secure Workload を使用して行う必要があります。
- ポリシーの適用が停止されるか、外部オーケストレータが削除されると、すべての Secure Workload ポリシーが *Citrix Netscaler* ロードバランサから削除されるため、ACL は空になります。

外部オーケストレータの OpenAPI ポリシー適用ステータスを使用して、外部オーケストレータに関連付けられたロード バランサ アプライアンスへの Secure Workload ポリシー適用のステータスを取得できます。この機能は、*Citrix Netscaler* アプライアンスへの ACL ルールの展開が成功したかどうかを確認するのに役立ちます。

警告

- 適用が有効になっている場合、Secure Workload ポリシーは常に ACL のグローバルリスト（パーティション *default*）に展開されます。
- 単一のアドレスとして指定された VIP のみがサポートされています。言い換えれば、アドレスパターンとして指定された VIP はサポートされていません。
- 検出されたサービス（*Citrix Netscaler* 仮想サーバー）の可視性はサポートされていません。

トラブルシューティング

- 接続の問題 Secure Workload は、Secure Workload アプライアンスサーバーの 1 つから、または TaaS の場合はクラウドから、または Secure Workload Secure Connector トンネル サービスをホストしている VM から、HTTPS 接続を使用して、指定された IP/ホスト名とポート番号に接続しようとします。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。また、指定されたクレデンシャルが正しいこと、および Citrix Netscaler アプライアンスに REST API 要求を送信するための読み取りおよび書き込みアクセス権限があることを確認してください。
- ACL ルールが見つからない ACL ルールが見つからない場合は、ポリシー適用の実行後に、対応する仮想サーバーが有効になっていることを確認してください。そのステータスは稼働状態である必要があります。

TAXII

TAXII（Trusted Automated Exchange of Intelligence Information）統合により、Secure Workload はセキュリティベンダーからの脅威インテリジェンス データ フィードを取り込み、ネットワー

クフローに注釈を付け、悪意のある IP、悪意のあるハッシュなどの STIX（構造化脅威情報表現）インジケータを使用してハッシュを処理できます。

「taxii」タイプの外部オーケストレータ構成が追加されると、Secure Workload アプライアンスは TAXII サーバーへの接続を試み、STIX データフィードコレクションをポーリングします。STIX データフィード（IP とバイナリ ハッシュ インジケータのみ）は解析され、ネットワークフローに注釈を付け、Secure Workload パイプライン内のハッシュを（オーケストレータが構成されているテナントに属するものとして）処理するために使用されます。

インポートされた悪意のある IP と一致するプロバイダーまたはコンシューマのアドレスを持つネットワークフローは、複数値ラベル「orchestrator_malicious_ip_by_<vendor name>」でタグ付けされます。<vendor name> は、ユーザーのオーケストレータ構成入力 TAXII ベンダーであり、ラベル値は「Yes」です。

取り込まれた STIX バイナリ ハッシュ インジケータは、ワークロード プロセス ハッシュに注釈を付けるために使用されます。これは、セキュリティダッシュボード/プロセスハッシュスコアの詳細、およびワークロードプロファイル/ファイルハッシュに表示されます（一致する場合）。

前提条件

- セキュアコネクタトンネル（接続に必要な場合）
- サポートされる TAXII サーバー：1.0
- STIX バージョンでサポートされる TAXII フィード：1.x

設定フィールド

「外部オーケストレータの作成」で説明されている共通の設定フィールドのほかに、次のフィールドを設定できます。

共通フィールド	必須	[説明（Description）]
[名前（Name）]	○	ユーザーが指定したオーケストレーションの名前。
[説明（Description）]	○	オーケストレーションのユーザー指定の説明。
[ベンダー（Vendor）]	○	ベンダーはインテリジェンス データ フィードを提供します。
[フルスナップショット間隔（Full Snapshot Interval）]	○	TAXII フィードの完全なスナップショットを実行する間隔（秒単位）。 （デフォルト：1 日）

共通フィールド	必須	[説明（Description）]
[ポーリングURL（Poll Url）]	○	ポーリングデータへのポーリングの完全な URL パス。
[コレクション（Collection）]	○	ポーリングされる TAXII フィードコレクション名。
[ポーリング日（Poll Days）]	○	TAXII フィードからポーリングする過去の脅威データの数。
[ユーザー名（Username）]		認証用のユーザー名。
[パスワード（Password）]		認証用のパスワード。
[証明書（Certificate）]		認証に使用されるクライアント証明書。
[キー（Key）]		クライアント証明書に対応するキー。
[CA証明書（CA Certificate）]		オーケストレーション エンドポイントを検証する CA 証明書。
[自己署名証明書の受け入れ（Accept Self-signed Cert）]		TAXII API サーバー証明書の厳密な SSL チェックを無効にするチェックボックス にするチェックボックス
[セキュアコネクタトンネル（Secureconnector Tunnel）]		セキュアコネクタトンネルを介したこのオーケストレータのホストへのトンネル接続 へのトンネル接続。
[ホストリスト（Hosts List）]	○	TAXII サーバーを指すホスト名/IP とポートのペア。

ワークフロー

- ユーザーは Secure Workload クラスタから該当 IP/ポートで TAXII サーバーに到達できることを最初に確認する必要があります。
- ポーリングパスと TAXII フィード名を使用して、正しい TAXII サーバーを設定します。

生成されたラベル

キー	値
orchestrator_system/orch_type	*TAXII*
orchestrator_system/cluster_id	Cisco Secure Workload クラスタ設定の UUID。
orchestrator_system/cluster_name	このクラスタの設定に付けられた名前。
orchestrator_malicious_ip_by_<vendor>	フロープロバイダー/コンシューマアドレスがインポートされた TAXII の悪意のある IP データと一致する場合、「Yes」。

警告

- TAXII 統合は、オンプレミスの Cisco Secure Workload のみでサポートされます。
- TAXII フィードからの IP およびハッシュインジケータのみが取り込まれます。
- 取り込まれる IP の最大数は、TAXII フィードあたり 100K（最近更新された数値）です。
- 取り込まれるハッシュの最大数は、すべての TAXII フィードで 500K（最近更新された数値）です。
- STIX バージョン 1.x の TAXII フィードのみがサポートされています。

トラブルシューティング

- 接続の問題

  Secure Workload は、Secure Workload アプライアンスサーバーの 1 つ、または Secure Workload Secure Connector VPN トンネルサービスをホストしている VM から、指定されたポーリング URL パスへの接続を試みます。この接続を正しく確立するには、このトラフィックを許可するようにファイアウォールを構成する必要があります。

TAXII オーケストレータのフルポーリングの動作

デフォルトのフルスナップショット間隔は 24 時間です。

フルスナップショット間隔ごとに、Secure Workload は IP とハッシュの TAXII フィードを前述の制限までラベルデータベースにプルします。

Cisco Secure Firewall Management Center

Secure Workload と Cisco Secure Firewall Threat Defense のそれぞれの能力を組み合わせることで、特に次のような用途に役立つセキュリティソリューションを実現できます。

- ソフトウェアエージェントをインストールできないワークロードをセグメント化する。

  たとえば、ワークロードのオペレーティングシステム（アプライアンスベースのソフトウェア）を制御できない場合や、エージェントがサポートしていないレガシー オペレーティング システムでワークロードを実行している場合は、この統合を使用します。

- データセンターやクラウド内のさまざまなゾーンのトラフィックをセグメント化する。

  たとえば、ネットワークに入るトラフィック、ネットワークから出るトラフィック、およびネットワーク内のワークロード間のトラフィックに対して、さまざまなポリシーセットを簡単かつ広く適用できます。

この統合では、Secure Workload アプリケーション ワークスペースでセグメンテーションポリシーを作成します。適用されるポリシーが Secure Workload によって Cisco Secure Firewall Management Center 内のアクセスコントロールルールに変換されます。

ネットワークインベントリは、セグメンテーションポリシーのベースとなる Secure Workload インベントリフィルタによって動的に管理されます。ネットワークでワークロードが追加、変更、または削除されると、Cisco Secure Firewall Management Center 内のダイナミックオブジェクトが Secure Workload により自動的に更新されます。 対応するアクセスコントロールルールは、これらのダイナミックオブジェクトに基づきます。インベントリの変更および適用されたポリシーの変更はすべて、管理対象の Cisco Secure Firewall Threat Defense（旧称 Firepower Threat Defense、FTD）デバイスに自動的に展開されます。Secure Firewall Management Center で変更を再展開する必要はありません。

動作の詳細、サポートされているプラットフォーム、制限、両方の製品のセットアップ手順、トラブルシューティング情報など、この統合に関する完全な情報については、「Cisco Secure Workload および Cisco Secure Firewall Management Center 統合ガイド」を参照してください。

オーケストレータにより生成されるラベル

なし：Cisco Secure Firewall Management Center 外部オーケストレータはユーザー注釈を生成しません。

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。