OpenAPI

OpenAPI は、Secure Workload 機能用の REST API を提供します。

- OpenAPI 認証 （2 ページ）
- ワークスペースとセキュリティポリシー （4 ページ）
- スコープ （60 ページ）
- ロール （66 ページ）
- ユーザ（Users） （71 ページ）
- インベントリ フィルタ （77 ページ）
- フロー検索 （81 ページ）
- インベントリ （90 ページ）
- ワークロード （96 ページ）
- 施行 （107 ページ）
- クライアントサーバー構成 （115 ページ）
- ソフトウェアエージェント （121 ページ）
- Cisco Secure Workload ソフトウェアのダウンロード （129 ページ）
- Cisco Secure Workload エージェントのアップグレード （132 ページ）
- スイッチ （133 ページ）
- 収集ルール （135 ページ）
- ユーザーがアップロードしたファイルハッシュ （138 ページ）
- ユーザー定義ラベル （140 ページ）
- Virtual Routing and Forwarding（VRF） （151 ページ）
- オーケストレーション （155 ページ）
- オーケストレータのゴールデンルール （163 ページ）
- FMC オーケストレータドメイン （165 ページ）
- RBAC（役割ベースのアクセス制御）に関する考慮事項 （168 ページ）
- 高可用性とフェールオーバーに関する考慮事項 （168 ページ）
- Kubernetes RBAC リソースに関する考慮事項 （168 ページ）
- サイト情報 （170 ページ）
- クラスタの正常性 （171 ページ）
- Service Health （171 ページ）

• Secure Connector （172 ページ）
• 外部オーケストレータのポリシー適用ステータス （173 ページ）
• 管理対象データタップとデータシンクの証明書のダウンロード （174 ページ）
• 変更ログ （176 ページ）
• ルーティング不可能なエンドポイント （179 ページ）

OpenAPI 認証

OpenAPI はダイジェスト ベースの認証方式を使用します。ワークフローは次のようになります。

1. Secure Workload UI ダッシュボードにログインします。
2. 目的の機能を使用して API キーと API 秘密を生成します。
3. JSON 形式で REST 要求を送信するには、Secure Workload API SDK を使用します。
4. Python SDK を使用するには、`pip install tetpyclient` を使用して SDK をインストールします。
5. Python sdk をインストールしたら、RestClient をインスタンス化するためのいくつかの定型コードを次に示します。

```
from tetpyclient import RestClient

API_ENDPOINT="https://<UI_VIP_OR_DNS_FOR_TETRATION_DASHBOARD>"

# ``verify`` は、SSL サーバー認証を無効にするオプションのパラメータです。
# デフォルトでは、クラスタダッシュボード IP は、自己署名された証明書を使用します。
# 展開。そのため、「verify = False」を使用してサーバを無効にすることができます。
# API クライアントの SSL での認証。ユーザーが自身をアップロードした場合
# クラスタへの証明書（[プラットフォーム（Platform）] > [SSL証明書（SSL Certificate）] から）
# エンタープライズ CA によって署名された後、サーバー側の認証
# 有効にする必要があります。このようなシナリオでは、コード verify=False
# は verify="path-to-CA-file" に置き換える必要があります。
# credentials.json は次のようになります。
# {
#"api_key": "<hex string>",
#"api_secret": "<hex string>"
# }
restclient = RestClient(API_ENDPOINT,
credentials_file='<path_to_credentials_file>/credentials.json',
verify=False)
```

```
# API 呼び出しの後に、エージェントのリストを取得する API などがあります。
# API は passed /openapi/v1/sensors または just /sensors です。
resp = restclient.get('/roles')
```

API キーと秘密の生成

ステップ 1 Secure Workload のWeb インターフェイスで、ウィンドウの右上隅にある人のアイコンをクリックし、[API キー（API Keys）] を選択します。

ステップ 2 [API キーの作成（Create API Key）] をクリックします。

ステップ 3 キーと秘密に必要な機能を指定します。ユーザーは、API キー + 秘密のペアの使用を予定している機能の制限設定を選択する必要があります。ユーザーが使用できる API 機能は、ユーザーのロールによって異なります。たとえば、サイト管理者ユーザーは、ソフトウェア エージェントを管理するためのキーを生成できますが、この機能は非サイト管理者ユーザーには使用できません。

API 機能には次のものがあります。

- SW エージェント管理（`sensor_management`）：SW エージェントのステータスを設定およびモニタリングできます
- Cisco Secure Workload ソフトウェアのダウンロード（`software_download`）：Secure Workload エージェント/仮想アプライアンスのソフトウェアパッケージをダウンロードできます
- フローとインベントリ検索（`flow_inventory_query`）：Secure Workload クラスタ内のフローとインベントリ項目を照会できます
- ユーザー、ロール、範囲の管理（`user_role_scope_management`）：ユーザー、ロール、範囲を読み取り/追加/変更/削除できます
- ユーザーデータのアップロード（`user_data_upload`）：ユーザーがフローとインベントリ項目に注釈を付けるためにデータをアップロードしたり、正当または不当なファイルハッシュをアップロードしたりできます
- ワークスペースとポリシー管理（`app_policy_management`）：ワークスペース（「アプリケーション」）を管理し、ポリシーを適用できます
- 外部システム統合: vCenter、Kubernetes などの外部システムと統合できます
- Cisco Secure Workload アプライアンス管理：Secure Workload クラスタを管理できます（サイト管理ユーザーのみが利用可能）

ステップ 4 [作成（Create）] をクリックします。

ステップ 5 キーと秘密をコピーして貼り付け、安全な場所に保存します。または、API クレデンシャル ファイルをダウンロードします。

（注） LDAP による外部認証や LDAP 認証が有効になっている場合、ユーザーセッションが終了するとグループの LDAP メンバーから派生した Secure Workload ロールが再評価されるため、API キーを介した OpenAPI へのアクセスはシームレスに機能しなくなります。したがって、中断のない OpenAPI アクセスを保証するために、API キーを持つすべてのユーザーが [ユーザー詳細の編集（Edit User Details）] フローで [ローカル認証を使用（Use Local Authentication）] オプションを有効にすることを推奨します。

ワークスペースとセキュリティポリシー

次のページでは、セグメンテーションを管理するための OpenAPI エンドポイントについて説明します。

ワークスペース

ワークスペース（以前の呼称は「アプリケーションワークスペース」または「アプリケーション」）は、特定の範囲のポリシーを定義、分析、および適用するためのコンテナです。動作の詳細については、『ワークスペース』のマニュアルを参照してください。この一連の API には、API キーに関連付けられている `app_policy_management` 機能が必要です。

ワークスペースオブジェクト

ワークスペース（「アプリケーション」）の JSON オブジェクトは、API エンドポイントに応じて、単一のオブジェクトまたはオブジェクトの配列として返されます。オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	ワークスペースの一意の識別子。
name	string	ユーザーが指定したワークスペース名。
description	string	ユーザーが指定したワークスペースの説明。
app_scope_id	string	ワークスペースが関連付けられている範囲の ID。
author	string	ワークスペースを作成したユーザーの姓と名。
primary	boolean	ワークスペースがその範囲に対してプライマリであるかどうかを示します。

属性	タイプ	説明
alternate_query_mode	boolean	ワークスペースに「ダイナミックモード」が使用されているかどうかを示します。ダイナミックモードでは、自動ポリシー検出の実行により、クラスタごとに1つ以上の候補クエリが作成されます。デフォルト値は true です。
created_At	整数	ワークスペースが作成されたときの Unix タイムスタンプ。
latest_adm_version	integer	ワークスペースの最新の adm(v*) バージョン。
analysis_enabled	boolean	ワークスペースで分析が有効になっているかどうかを示します。
analyzed_version	integer	ワークスペースの分析された p* バージョン。
enforcement_enabled	boolean	ワークスペースで適用が有効になっているかどうかを示します。
enforced_version	integer	ワークスペースの適用された p* バージョン。

アプリケーションの一覧表示

このエンドポイントは、ワークスペース（「アプリケーション」）の配列を返します。

```
GET /openapi/v1/applications
```

表 *1:* パラメータ

名前	タイプ	説明
app_scope_id	string	特定のアプリケーション範囲に関連付けられたワークスペースを照合します。
exact_name	string	ワークスペースを提供された値と正確に照合します。

応答オブジェクト：ワークスペースオブジェクトの配列を返します。

サンプル python コード

```
restclient.get('/applications')
```

単一のワークスペースの取得

このエンドポイントは、要求されたワークスペース（「アプリケーション」）を 1 つの JSON オブジェクトとして返します。

```
GET /openapi/v1/applications/{application_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

応答オブジェクト：指定された ID のワークスペースオブジェクトを返します。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
restclient.get('/applications/%s' % application_id)
```

ワークスペースの作成

このエンドポイントは、ワークスペース（「アプリケーション」）を作成します。クラスタとポリシーの定義を含む JSON 本文をポストすることによって、ポリシーを定義することができます。

（注） 同じ範囲にプライマリワークスペースが存在し、新しいポリシーが提供されている場合、ポリシーは新しいバージョンとして既存のワークスペースに追加されます。

```
POST /openapi/v1/applications
```

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
app_scope_id	string	ワークスペースに割り当てる範囲 ID。
name	string	（オプション）ワークスペースの名前。
説明	string	（オプション）ワークスペースの説明。

名前	タイプ	説明
alternate_query_mode	boolean	（オプション）ワークスペースに「ダイナミックモード」が使用されているかどうかを示します。ダイナミックモードでは、自動ポリシー検出の実行により、クラスタごとに1つ以上の候補クエリが作成されます。デフォルト値は true です。
strict_validation	boolean	（オプション）アップロードされたデータに不明なキー/属性がある場合、エラーを返します。スペルミスされたキーを検出するのに役立ちます。デフォルト値は false です。
プライマリ	string	（オプション）このワークスペースを関連付けられた範囲のプライマリにする必要がある場合は、「true」に設定します。**デフォルトは true です。**

ワークスペース内で作成されるポリシーを記述する、追加のオプションパラメータを含めることもできます。

（注） このスキームは、UI と**詳細**エンドポイントからのエクスポート時に返されるものに対応しています。

名前	タイプ	説明
クラスタ	クラスタの配列	ポリシーを定義するために使用されるノードのグループ。
inventory_filters	インベントリ フィルタの配列	データセンター アセットをフィルタリングします。
absolute_policies	ポリシーの配列	絶対ランクで作成される順序ポリシー。
default_policies	ポリシーの配列	デフォルト ランクで作成される順序ポリシー。
catch_all_action	string	「ALLOW」または「DENY」

クラスタ オブジェクト属性:

名前	タイプ	説明
id	string	ポリシーで使用される固有識別子。
name	string	クラスタの表示名。
説明	string	クラスタの説明。
nodes	ノードの配列	このクラスタに属するノードまたはエンドポイント 。
consistent_uuid	string	特定のワークスペースに対して一意である必要があります。自動ポリシー検出の実行後、類似/同じクラスタは次のバージョンでも consistent_uuid を維持します。

ノード オブジェクトの属性:

名前	タイプ	説明
ip	string	ノードの IP またはサブネット。例：10.0.0.0/8 または 1.2.3.4
name	string	ノードの表示名。

インベントリ フィルタ オブジェクトの属性:

名前	タイプ	説明
id	string	ポリシーで使用される固有識別子。
name	string	クラスタの表示名。
query	object	インベントリ フィルタ クエリの JSON オブジェクト表現。

ポリシー オブジェクトの属性:

名前	タイプ	説明
consumer_filter_id	string	クラスタの ID、ユーザー インベントリ フィルタ、またはアプリ範囲。
provider_filter_id	string	クラスタの ID、ユーザー インベントリ フィルタ、またはアプリ範囲。

名前	タイプ	説明
action	string	「ALLOW」または「DENY」
l4_params	l4params の配列	許可されたポートとプロトコルのリスト。

L4Params オブジェクト属性：

名前	タイプ	説明
proto	整数	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
port	array	ポートの包含範囲。例：[80, 80] または [5000, 6000]。
approved	boolean	（オプション）ポリシーが承認されているかどうかを示します。デフォルトは False です。

応答オブジェクト：新しく作成されたワークスペースオブジェクトを返します。

サンプル python コード

```
name = 'test'
scope_id = '5ce480cc497d4f1b4b9a9e8d'
filter_id = '5ce480cd497d4f1b4b9a9ea4'
application = {
'app_scope_id': scope_id,
'name': name,
'absolute_policies': [
{
# consumer/provider filter IDs can be ID of a cluster identified during
‹→automatic policy discovery (formerly known as ADM),
# user inventory filter or app scope.
'provider_filter_id': filter_id,
'consumer_filter_id': filter_id,
'action': 'ALLOW',
# ALLOW policy for TCP on port 80.
'l4_params': [
{
'proto': 6, # TCP
'port': [80, 80], # port range
}
],
}
],
'catch_all_action': 'ALLOW'
}
restclient.post('/applications', json_body=json.dumps(application))
```

新しいバージョンのインポート

ポリシーをインポートし、ワークスペース（「アプリケーション」）の新しい v* バージョンを作成します。

```
POST /openapi/v1/applications/{application_id}/import
```

パラメータは、上記のワークスペース エンドポイントの作成と同じです。

応答オブジェクト：ワークスペースオブジェクトを返します。

一連のポリシーの検証

新しいバージョンを作成せずに一連のポリシーを検証します。

```
POST /openapi/v1/applications/validate_policies
```

app_scope_idは必須です。残りのパラメータは、前述のワークスペースエンドポイントの作成と同じです。

応答オブジェクト：

属性	タイプ	説明
有効な	boolean	ポリシーが有効かどうかを示します。
errors	アレイ	無効な場合、エラーの詳細。

ワークスペースの削除

ワークスペース（「アプリケーション」）を削除します。

```
DELETE /openapi/v1/applications/{application_id}
```

ワークスペースを削除する前に、ワークスペースで適用を無効にする必要があります。

ワークスペースまたはそのクラスタが他のアプリケーションによって（提供されるサービスの関係を介して）使用されている場合、このエンドポイントは`422 Unprocessable Entity`を返します。返されるエラーオブジェクトには、`details`属性が含まれ、依存オブジェクトの数とともに各タイプの最初の10個のIDが示されます。この情報を使用して、問題となっている依存関係を見つけて削除できます。

パラメータ：要求URLには、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

応答オブジェクト：なし

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
restclient.delete('/applications/%s' % application_id)
```

ワークスペースの更新

このエンドポイントにより、既存のワークスペース（「アプリケーション」）が更新されます。

```
PUT /openapi/v1/applications/{application_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
name	string	（オプション）ワークスペースの更新された名前。
descrip	string	（オプション）ワークスペースの更新された説明。
プライマリ	string	（オプション）ワークスペースをプライマリにするには、「true」に設定します。ワークスペースをセカンダリにするには、「false」に設定します。

応答オブジェクト：指定された ID の更新されたワークスペースオブジェクト。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
req_payload = {
'name': 'Updated Name',
'description': 'Updated Description',
'primary': 'true'
}
resp = restclient.put('/applications/%s' % application_id,
json_body=json.dumps(req_payload))
```

ワークスペースの詳細を取得

このエンドポイントは、ワークスペースの完全なエクスポート JSON ファイルを返します。これには、ポリシーとクラスタの定義が含まれます。

```
GET/openapi/v1/applications/{application_id}/details
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
version	string	（オプション）「v10」または「p10」形式のバージョンで、デフォルトは「最新」です。

応答オブジェクト：指定されたワークスペースバージョンのクラスタとポリシーを返します。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
# For v* version v10 and for p* version p10
version = 'v10'
resp = restclient.get('/applications/%s/details?version=%s' % (application_id,
‹→version))
```

ワークスペースバージョンの一覧表示

このエンドポイントは、特定のワークスペースのすべてのバージョンのリストを返します。

```
GET /openapi/v1/applications/{application_id}/versions
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
created_before	integer	（オプション）ページネーションの場合、前の応答からの最新バージョンの「created_at」（作成日）に設定します。
limit	integer	（オプション）返される結果の最大数。デフォルトは 50 です。

応答オブジェクト：次の属性を持つオブジェクトの配列：

属性	タイプ	説明
version	string	「v10」または「p10」の形式のバージョン。

属性	タイプ	説明
created_At	整数	ワークスペースが作成されたときのUnixタイムスタンプ。
説明	string	ユーザーが提供する説明。
name	string	表示名。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'

created_before = 1612325705

limit = 10

resp = restclient.get('/applications/%s/versions?created_before=%s&limit=%s' %

(application_id, created_before, limit))
```

ワークスペースバージョンの削除

このエンドポイントは、クラスタとポリシーを含む指定されたバージョンを削除します。適用または分析されたバージョンは削除されません。メンバーが外部ポリシーを介して別のワークスペースによって参照されている場合、応答は参照のリストと共にエラーを返します。

```
DELETE /openapi/v1/applications/{application_id}/versions/{version}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
version	string	「v10」または「p10」の形式のバージョン。

応答オブジェクト：なし

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'

version = 'v10'

resp = restclient.delete('/applications/%s/versions/%s' %

(application_id, version))
```

ワークスペースバージョンの比較

このエンドポイントは、提供されたワークスペースバージョン間の相違を計算します。追加されたポリシー、削除されたポリシー、およびオプションで変更されていないポリシーを返しま

す。両方のバージョンに、一致する consistent_uuid によって定義されたクラスタが存在し、クエリが変更された場合、クラスタの変更が含まれます。

```
GET /openapi/v1/applications/{application_id}/version_diff
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
base_version	string	「v10」または「p10」などの完全バージョン。
draft_version	string	「v10」または「p10」などの完全バージョン。
include_unchanged	boolean	デフォルトは false です。変更されていないポリシーを応答で返します。

応答オブジェクト：次の属性を持つオブジェクトが返されます。

属性	タイプ	説明
クラスタ	アレイ	バージョン間で変更されたクラスター。
ポリシー	アレイ	バージョン間で変更されたポリシー。

最新のポリシーの分析

ワークスペース内で一連の最新のポリシー分析を有効にします。

```
POST /openapi/v1/applications/{application_id}/enable_analysis
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

パラメータ：オプションの JSON クエリ本体には、次のキーが含まれます

名前	タイプ	説明
action_note	string	（オプション） ポリシー公開アクションの理由。

名前	タイプ	説明
name	string	（オプション） 公開されたポリシーバージョンの名前。
説明	string	（オプション） 公開されたポリシーバージョンの説明。

応答オブジェクト：次の属性を持つオブジェクトが返されます。

属性	タイプ	説明
data_set	オブジェクト	データセットのJSONオブジェクト表現。
analyzed_policy_version	integer	ワークスペースの分析されたp* バージョン。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
req_payload = {
'action_note': 'Policy analysis',
'name': 'Test run 1',
'description': 'New workloads added.'
}
resp = restclient.post('/applications/%s/enable_analysis' % application_id,
json_body=json.dumps(req_payload))
```

単一のワークスペースでポリシー分析を無効にする

ワークスペースでポリシー分析を無効にします。

```
POST /openapi/v1/applications/{application_id}/disable_analysis
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

応答オブジェクト：次の属性を持つオブジェクトが返されます。

属性	タイプ	説明
data_set	オブジェクト	データセットのJSONオブジェクト表現。

analyzed_policy_version	integer	最後に分析されたワークスペースの p* バージョン。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'

resp = restclient.post('/applications/%s/disable_analysis' % application_id)
```

単一のワークスペースの適用

ワークスペースで最新のポリシーセットの適用を有効にします。

```
POST /openapi/v1/applications/{application_id}/enable_enforce
```

警告 新しいホスト ファイアウォール ルールが挿入され、関連するホスト上で既存のルールがすべて削除されます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
version	string	(オプション)適用するポリシーのバージョン。

バージョンが指定されていない場合、ワークスペースの最新のポリシーが適用されます。バージョンは「p*」の形式で指定することが推奨されます。整数のみが指定されている場合、対応する「p*」バージョンが適用されます。

応答オブジェクト：次の属性を持つオブジェクトが返されます。

名前	タイプ	説明
新時代	string	最新の適用プロファイルの固有識別子。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'

req_payload = {

'version': 'p10'

}

resp = restclient.post('/applications/%s/enable_enforce' % application_id,

json_body=json.dumps(req_payload))
```

単一ワークスペースの適用の無効化

ワークスペースへの適用を無効にします。

```
POST /openapi/v1/applications/{application_id}/disable_enforce
```

警告 新しいホスト ファイアウォール ルールが挿入され、関連するホスト上で既存のルールがすべて削除されます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

応答オブジェクト：次の属性を持つオブジェクトを返します。

名前	タイプ	説明
新時代	string	最新の適用プロファイルの固有識別子。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'

resp = restclient.post('/applications/%s/disable_enforce' %

application_id)
```

自動ポリシー検出の開始

ワークスペースのポリシーを自動的に検出します（以前は「ADM 実行の送信」と呼ばれていました）。

```
POST /openapi/v1/applications/{application_id}/submit_run
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
start_time	string	自動ポリシー検出実行の入力時間間隔の開始時刻。

名前	タイプ	説明
end_time	string	自動ポリシー検出実行の入力時間間隔の終了時刻。
clustering_granularity	string	（オプション）クラスタリングの粒度により、ユーザーは自動ポリシー検出によって生成されるクラスタのサイズを制御できます。想定される値：VERY_FINE、FINE、MEDIUM、COARSE、またはVERY_COARSE
port_generalization	string	（オプション）ポートの汎用化により、ポートの汎用化を実行するときに必要な統計的有意性のレベルが制御されます。想定される値：DISABLED、CONSERVATIVE、MODERATE、AGGRESSIVE、またはVERY_AGGRESSIVE
policy_compression	string	（オプション）ポリシー圧縮が有効になっている場合、ワークスペース内に生成されたクラスタの中で、相当な頻度で使用されるポリシー（同じプロバイダーポートを使用するポリシー）は、親に「除外」される可能性があります。つまり、親範囲全体に置き換えられる可能性があります。想定される値：DISABLED、CONSERVATIVE、MODERATE、AGGRESSIVE、またはVERY_AGGRESSIVE
auto_accept_policy_connectors	boolean	（オプション）自動受け入れポリシーコネクタ 自動ポリシー検出中に作成されたすべての発信ポリシー要求は、自動的に受け入れられます。

名前	タイプ	説明
enable_exclusion_filter	boolean	（オプション）除外フィルタの有効化オプションは、ユーザー定義の除外フィルター（存在する場合）のいずれかに一致するすべてのカンバセーションを無視する柔軟な対応を可能にします。詳細については、「除外フィルタ」を参照してください。
enable_default_exclusion_filter	boolean	（オプション）デフォルトの除外フィルタの有効化オプションは、デフォルトの除外フィルタ（存在する場合）のいずれかに一致するすべてのカンバセーションを無視する柔軟な対応を可能にします。詳細については、「デフォルトの除外フィルタ」を参照してください。
enable_service_discovery	boolean	（オプション）エージェントでのサービス検出の有効化が設定されている場合、エージェントノードに存在するサービスに関連した一時的なポート範囲情報がレポートされます。次に、レポートされたポート範囲情報に基づいてポリシーが生成されます。
carry_over_policies	boolean	（オプション）承認済みポリシーの引継ぎが設定されている場合、UI または OpenAPI を使用してユーザーによって承認済みとしてマークされているすべてのポリシーが保持されます。

名前	タイプ	説明
skip_clustering	boolean	（オプション）クラスタリングのスキップが設定されている場合、新しいクラスタは生成されず、既存の承認済みクラスタまたはインベントリフィルタからポリシーが生成されます。設定されていない場合は、範囲内のすべてのワークロードがポリシーに含められます。
deep_policy_generation	boolean	（オプション）ディープポリシー生成は、グローバルポリシー生成に関心がある場合に特に役立ちます。詳細については、「ディープポリシー生成」をご覧ください。
use_default_config	boolean	（オプション）このオプションが設定されている場合、自動ポリシー検出は、以前の実行設定の代わりにデフォルトのポリシー検出設定を使用します。詳細については、「デフォルトのポリシー検出設定」を参照してください。

（注） ワークスペースで自動ポリシー検出が以前に実行された場合、指定されていないオプションパラメータのデフォルト値は、以前の自動ポリシー検出実行設定から取得されます。それ以外の場合、デフォルト値はデフォルトのポリシー検出設定から取得されます。

応答オブジェクト：次の属性を持つオブジェクトが返されます。

名前	タイプ	説明
message	string	自動ポリシー検出実行の成功/失敗に関するメッセージ。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'

req_payload = {

'start_time': '2020-09-17T10:00:00-0700',
```

```
'end_time': '2020-09-17T11:00:00-0700',
# Optional Parameters.
'clustering_granularity': 'FINE',
'port_generalization': 'AGGRESSIVE',
'policy_compression': 'AGGRESSIVE',
'auto_accept_policy_connectors': False,
'enable_exclusion_filter': True,
'enable_default_exclusion_filter': True,
'enable_service_discovery': True,
'carry_over_policies': True,
'skip_clustering': False,
'deep_policy_generation': True,
'use_default_config': False
}
resp = restclient.post('/applications/%s/submit_run' % application_id,
json_body=json.dumps(req_payload))
```

ポリシー検出実行のステータスの取得

ワークスペースで実行されている自動ポリシー検出のステータスをクエリします。

```
GET /openapi/v1/applications/{application_id}/adm_run_status
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの固有識別子。

応答オブジェクト：次の属性を持つオブジェクトを返します。

名前	タイプ	説明
status	string	自動ポリシー検出実行のステータス。値：PENDING、COMPLETE、または FAILED

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'
resp = restclient.get('/applications/%s/adm_run_status' % application_id)
```

ポリシー

この一連の API を使用して、ポリシーの追加、編集、または削除を管理できます。 create および update catch all アクションには `version` パラメータが必要です。API キーに関連付けられた `user_role_scope_management` 機能が必要です。

ポリシー オブジェクト

ポリシーオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	ポリシーの一意の識別子。
application_id	string	ポリシーが属するワークスペースの ID。
consumer_filter_id	string	定義されたフィルタの ID。現在、任意のクラスタ、ユーザー定義フィルタ、または範囲をポリシーのコンシューマとして使用できます。
provider_filter_id	string	定義されたフィルタの ID。現在、任意のクラスタ、ユーザー定義フィルタ、または範囲をポリシーのプロバイダーとして使用できます。
version	string	ポリシーが属するワークスペースのバージョンを示します。
rank	string	ポリシーランク。入力可能値：DEFAULT、ABSOLUTE、CATCHALL。
policy_action	string	入力可能値は、ALLOW または DENY です。コンシューマとプロバイダー間の特定のサービスポートまたはプロトコルについて、トラフィックを許可するかドロップするかを示します。
priority	integer	ポリシーを並べ替えるために使用されます。

属性	タイプ	説明
l4_params	l4params の配列	許可されたポートとプロトコルのリスト。

L4Params オブジェクト属性：

名前	タイプ	説明
proto	整数	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
port	array	ポートの包含範囲。例：[80, 80] または [5000, 6000]。
description	string	このプロトとポートに関する短い文字列。
approved	boolean	ポリシーがユーザーによって承認されている場合。

ポリシーの取得

このエンドポイントは、特定のワークスペースのポリシーのリストを返します。このAPIは、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/policies
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
version	string	ポリシーを取得するワークスペースのバージョンを示します。
consumer_filter_id	string	（オプション）コンシューマフィルタ ID で出力をフィルタ処理します。
provider_filter_id	string	（オプション）コンシューマフィルタ ID で出力をフィルタ処理します。

以下に示すように、この特定のワークスペース内のすべてのポリシーのオブジェクトを返します。

```
{

absolute_policies: [ ... ],
```

default_policies: [...],

```
catch_all_action:
```

```
}
```

サンプル python コード

```
application_id = '5f88c996755f023f3bafe163'
```

```
restclient.get('/applications/%s/policies' % application_id, params={'version': '1'})
```

デフォルトポリシーの取得

このエンドポイントは、特定のワークスペースのデフォルトポリシーのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/default_policies
```

パラメータ：

名前	タイプ	説明
id	string	ポリシーの固有識別子。
version	string	ポリシーを取得するワークスペースのバージョンを示します。
limit	inte-	要求あたりのポリシー数を制限します。
get offset inte	inte-	（オプション）以前の応答から受け取ったオフセット番号。常に `limit` と共に使用する必要があります。
consumer_filter_id	string	（オプション）コンシューマフィルタ ID で出力をフィルタ処理します。
provider_filter_id	string	（オプション）プロバイダーフィルタ ID で出力をフィルタ処理します。

このワークスペースの提供されたバージョンのデフォルトポリシーのリストを返します。応答には、要求された数のポリシーとオフセットが含まれており、次のポリシーのセットを取得するには、後続の要求でこのオフセットを使用します。応答にオフセットがない場合は、すべてのポリシーが既に取得されていることを示します。

サンプル python コード

```
application_id = '5f88c996755f023f3bafe163'
```

```
restclient.get('/applications/%s/default_policies' % application_id, params={'version
→': '1', 'limit': 3, 'offset': 3})
```

サンプル応答

```
{
"results": [
PolicyObject4,
PolicyObject5,
PolicyObject6
],
"offset": 6
}
```

絶対ポリシーの取得

このエンドポイントは、特定のワークスペース内の絶対ポリシーのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/absolute_policies
```

パラメータ：

名前	タイプ	説明
version	string	ポリシーを取得するワークスペースのバージョンを示します。
limit	integer	要求あたりのポリシー数を制限します。
offset	integer	（オプション）以前の応答から受け取ったオフセット番号。常に `limit` と共に使用する必要があります。
consumer_filter_id	string	（オプション）コンシューマフィルタ ID で出力をフィルタ処理します。
provider_filter_id	string	（オプション）プロバイダーフィルタ ID で出力をフィルタ処理します。

このワークスペースの提供されたバージョン内の絶対ポリシーのリストを返します。応答には、要求された数のポリシーとオフセットが含まれており、次のポリシーのセットを取得する

には、後続の要求でこのオフセットを使用します。応答にオフセットがない場合は、すべてのポリシーが既に取得されていることを示します。

サンプル python コード

```
application_id = '5f88c996755f023f3bafe163'

restclient.get('/applications/%s/absolute_policies' % application_id, params={'version
→': '1', 'limit': 3})
```

サンプル応答

```
{

"results": [

PolicyObject1,

PolicyObject2,

PolicyObject3

],

"offset": 3

}
```

Catch All ポリシーの取得

このエンドポイントは、特定のワークスペースの Catch All ポリシーを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/catch_all
```

パラメータ：

名前	タイプ	説明
version	string	ポリシーを取得するワークスペースのバージョンを示します。

ワークスペースの指定されたバージョンの Catch All ポリシーを表す単一のポリシーオブジェクトを返します。

サンプル python コード

```
application_id = '5f88c996755f023f3bafe163'

restclient.get('/applications/%s/catch_all' % application_id, params={'version': '1'})
```

特定のポリシーの取得

このエンドポイントは、ポリシーのインスタンスを返します。

```
GET /openapi/v1/policies/{policy_id}
```

指定した ID に関連付けられているポリシーオブジェクトを返します。

サンプル python コード

```
policy_id = '5f88ca1e755f0222f85ce85c'

restclient.get('/policies/%s' % policy_id)
```

ポリシーの作成

このエンドポイントは、新しいポリシーを作成するために使用されます。

```
POST /openapi/v1/applications/{application_id}/policies
```

パラメータ：

属性	タイプ	説明
consumer_filter_id	string	定義されたフィルタの ID。
provider_filter_id	string	定義されたフィルタの ID。
version	string	ポリシーを更新するワークスペースのバージョンを示します。
rank	string	値は、ランク付けの DEFAULT、ABSOLUTE、または CATCHALL になります。
policy_action	string	値は ALLOW または DENY です。つまり、指定されたサービスポートまたはプロトコルで、コンシューマからプロバイダーへのトラフィックを許可またはドロップします。
priority	integer	ポリシーを並べ替えるために使用されます。

サンプル python コード

```
req_payload = {

"version": "v1",

"rank" : "DEFAULT",

"policy_action" : "ALLOW",

"priority" : 100,

"consumer_filter_id" : "123456789",

"provider_filter_id" : "987654321",

}

resp = restclient.post('/openapi/v1/applications/{application_id}/policies', json_
```

```
,→body=json.dumps(req_payload))
```

デフォルトポリシーの作成

このエンドポイントは、新しいデフォルトポリシーを作成するために使用されます。このエンドポイントでは、ポリシーエンドポイントの作成と同様のデフォルトポリシーが作成されます。

```
POST /openapi/v1/applications/{application_id}/default_policies
```

絶対ポリシーの作成

このエンドポイントは、新しい絶対ポリシーを作成するために使用されます。このエンドポイントでは、ポリシーエンドポイントの作成と同様の絶対ポリシーが作成されます。

```
POST /openapi/v1/applications/{application_id}/absolute_policies
```

ポリシーの更新

このエンドポイントはポリシーを更新します。

```
PUT /openapi/v1/policies/{policy_id}
```

パラメータ：

属性	タイプ	説明
consumer_filter_id	string	定義されたフィルタの ID。
provider_filter_id	string	定義されたフィルタの ID。
policy_action	string	可能な値は、ALLOW または DENY です。コンシューマとプロバイダー間の特定のサービスポートまたはプロトコルについて、トラフィックを許可するかドロップするかを示します。
priority	integer	ポリシーの優先順位の並べ替えに使用

指定された ID に関連付けられている変更されたポリシーオブジェクトを返します。

Catch All の更新

このエンドポイントは、特定のワークスペースの Catch All を更新します。

```
PUT /openapi/v1/applications/{application_id}/catch_all
```

パラメータ：

属性	タイプ	説明
version	string	ポリシーを更新するワークスペースのバージョンを示します。
policy_action	string	可能な値は、ALLOW または DENY です。このワークスペースのいずれのポリシーにも一致しないトラフィックを許可するか、ドロップするかを示します。

ポリシーへのサービスポートの追加

このエンドポイントは、特定のポリシーのサービスポートを作成するために使用されます。

```
POST /openapi/v1/policies/{policy_id}/l4_params
```

パラメータ：

属性	タイプ	説明
version	string	ポリシーを取得するワークスペースのバージョンを示します。
start_port	integer	範囲の開始ポート。
end_port	integer	範囲の終了ポート。
proto	整数	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
説明	string	（オプション）このプロトコルとポートに関する短い文字列。

ポリシーのサービスポートの更新

このエンドポイントは、ポリシーの指定されたサービスポートを更新します。

```
PUT /openapi/v1/policies/{policy_id}/l4_params/{l4_params_id}
```

パラメータ：

属性	タイプ	説明
approved	bool	ポリシーを承認済みとしてマークします。

ポリシーのサービスポートの削除

このエンドポイントは、ポリシーの指定されたサービスポートを削除します。（オプション）詳細については、「除外フィルタ」を参照してください。

```
DELETE /openapi/v1/policies/{policy_id}/l4_params/{l4_params_id}
```

パラメータ：

属性	タイプ	説明
create_exclusion_filter	bool	（オプション）trueの場合、ポリシーに一致する除外フィルタが作成されます。このフィルタに一致するフローは、今後の自動ポリシー検出の実行から除外されます。詳細については、「除外フィルタ」を参照してください。

ポリシーの削除

このエンドポイントは、指定されたポリシーを削除します。除外フィルタは作成されません。

```
DELETE /openapi/v1/policies/{policy_id}
```

ポリシーの簡易分析

このエンドポイントを使用して、ルート範囲で分析または適用されたポリシーと一致する仮想フローの一連のポリシーを見つけることができます。詳細については、「簡易分析」を参照してください。

この API は、ルート範囲への最小限の読み取りアクセス権を持つユーザーのみが使用できます。また、API キーに関連付けられた `app_policy_management` 機能が必要です。

```
POST /openapi/v1/policies/{rootScopeID}/quick_analysis
```

クエリ本文は、次のスキーマを使用した JSON 本文で構成されます。

名前	タイプ	説明
consumer_ip	string	クライアント/コンシューマの IP アドレス。
provider_ip	string	サーバー/プロバイダーの IP アドレス。

名前	タイプ	説明
provider_port	integer	（オプション）プロバイダーポート。TCP または UDP フローにのみ関連します。
protocol	string	フローのプロトコル（TCP など）。
analysis_type	string	分析タイプは、**analyzed** または **enforced** にできます。分析タイプが「analyzed」の場合、ルート範囲で分析されたすべてのポリシーに対してフローを照合することによって、フローが決定されます。分析タイプが「enforced」の場合、ルート範囲で適用されたすべてのポリシーに対してフローを照合することによって、フローが決定されます。
application_id	string	（オプション）プライマリワークスペースの ID は常にワークスペース「v」バージョンを伴います。指定されている場合は、ルート範囲内の他のワークスペースからの分析および適用済みポリシーと共に、指定されたバージョンのポリシーを使用してフローが決定されます。このフィールドが省略された場合、ルート範囲内のすべての分析および適用済みポリシーを考慮して、フローが決定されます。
version	整数	（オプション）上記のワークスペースの「v」バージョン。これは、application_id が指定されている場合は指定する必要があります。それ以外の場合は省略する必要があります。

サンプル リクエスト

要求の本文は、JSON 形式のクエリである必要があります。

分析されたすべてのポリシーに基づいてフローが決定されるクエリ本文の例は、次のようになります。

```
req_payload = {

"consumer_ip": "4.4.1.1",

"provider_ip": "4.4.2.1",

"provider_port": 9081,

"protocol": "TCP",

"analysis_type": "analyzed"

}

resp = restclient.post('/openapi/v1/policies/{rootScopeID}/quick_analysis', json_

‹→body=json.dumps(req_payload))
```

フローの決定がワークスペースの「v」バージョンのポリシーと、ルート範囲内の他のすべてのワークスペースから分析されたポリシーに基づくクエリ本文の例は、次のようになります。

```
req_payload = {

"consumer_ip": "4.4.1.1",

"provider_ip": "4.4.2.1",

"provider_port": 9081,

"protocol": "TCP",

"analysis_type": "analyzed",

"application_id": "5e7e5f56497d4f0bc26c7bb3",

"version": 1

}

resp = restclient.post('/openapi/v1/policies/{rootScopeID}/quick_analysis', json_

‹→body=json.dumps(req_payload))
```

サンプル応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

キー	値
policy_decision	許可または拒否されるかどうかの仮想フローの決定。
outbound_policy	発信トラフィックを許可/拒否するコンシューマのポリシー
inbound_policy	着信トラフィックを許可/拒否するプロバイダーのポリシー

```
{
"policy_decision": "ALLOW",
"outbound_policy": {
"policy_rank": "DEFAULT",
"start_port": 9082,
"l4_detail_id": "5e7e600f497d4f7341f4f6d0",
"src_filter_id": "5e7e600e497d4f7341f4f459",
"end_port": 9082,
"cluster_edge_id": "5e7e600f497d4f7341f4f6d1",
"dst_filter_id": "5e7d0efc497d4f44b6b09351",
"action": "ALLOW",
"protocol": "TCP",
"app_scope_id": "5e7e5f3a497d4f0bc26c7bb0"
},
"inbound_policy": {
"policy_rank": "DEFAULT",
"start_port": 9082,
"l4_detail_id": "5e7e600f497d4f7341f4f6d0",
"src_filter_id": "5e7e600e497d4f7341f4f459",
"end_port": 9082,
"cluster_edge_id": "5e7e600f497d4f7341f4f6d1",
"dst_filter_id": "5e7d0efc497d4f44b6b09351",
"action": "ALLOW",
"protocol": "TCP",
"app_scope_id": "5e7e5f3a497d4f0bc26c7bb0"
}
}
```


ポリシーテンプレート

この一連の API は、ポリシーテンプレートを追加、編集、または削除するために使用できます。API キーに関連付けられている app_policy_management 機能が必要です。

ポリシーテンプレートの取得

このエンドポイントは、特定のルート範囲のポリシーテンプレートのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/application_templates?root_app_scope_id={root_app_scope_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_app_scope_id	string	ルート範囲の一意の識別子。

応答オブジェクト：指定されたルート範囲のポリシー テンプレート オブジェクトのリストを返します。

サンプル python コード

```
root_app_scope_id = '<root-app-scope-id>'

restclient.get('/application_templates?root_app_scope_id=%s' % root_app_scope_id)
```

特定のポリシーテンプレートの取得

このエンドポイントは、ポリシーテンプレートのインスタンスを返します。

```
GET /openapi/v1/application_templates/{template_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
template_id	string	ポリシーテンプレートの固有識別子。

応答オブジェクト：指定された ID のポリシー テンプレート オブジェクトを返します。

サンプル python コード

```
template_id = '<template-id>'

restclient.get('/application_templates/%s' % template_id)
```

ポリシーテンプレートの作成

このエンドポイントは、新しいポリシーテンプレートを作成するために使用されます。

```
POST /openapi/v1/application_templates
```

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
name	string	インポート時にテンプレートの名前として使用されます。
説明	string	（オプション）適用プロセス中に表示されるテンプレートの説明

属性	タイプ	説明
パラメータ	パラメータオブジェクト	テンプレートパラメータ（下記を参照）。
absolute_policies	ポリシーオブジェクトの配列	（オプション）絶対ポリシーの配列。
default_policies	ポリシーオブジェクトの配列	（必須）デフォルトポリシーの配列、空にすることが可能。

応答オブジェクト: 作成されたポリシーテンプレート オブジェクトを返します。

サンプル python コード

```
root_app_scope_id = '<root-app-scope-id>'

payload = {'root_app_scope_id': root_app_scope_id,

'name': "policy_name",

'default_policies': [

{

'action': 'ALLOW',

'priority': 100,

'l4_params': [

{

'proto': 17,

'port': [80, 90]

}

]

}

]

}

restclient.post('/application_templates',

json_body=json.dumps(payload))
```

ポリシーテンプレートの更新

このエンドポイントは、ポリシーテンプレートを更新します。

```
PUT /openapi/v1/application_templates/{template_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
template_id	string	ポリシーテンプレートの固有識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
name	string	（オプション）インポート時にテンプレートの名前として使用されます。
説明	string	（オプション）適用プロセス中に表示されるテンプレートの説明

応答オブジェクト：指定された ID の変更されたポリシーテンプレート オブジェクトを返します。

サンプル python コード

```
new_name = <new-name>

payload = {'name': new_name}

template_id = '<template-id>'

restclient.post('/application_templates/%s' % template_id,

json_body=json.dumps(payload))
```

ポリシーテンプレートの削除

このエンドポイントは、指定されたポリシーテンプレートを削除します。

```
DELETE /openapi/v1/application_templates/{template_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
template_id	string	プロファイルテンプレートの固有識別子。

応答オブジェクト：なし

サンプル python コード

```
template_id = '<template-id>'

restclient.delete('/application_templates/%s' % template_id)
```

ポリシーテンプレートのダウンロード

このエンドポイントは、ポリシーテンプレートをダウンロードします。

```
GET /openapi/v1/application_templates/{template_id}/download
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
template_id	string	ポリシーテンプレートの固有識別子。

応答オブジェクト：指定された ID を持つ完全なポリシーテンプレート定義を返します。

サンプル python コード

```
template_id = '<template-id>'

restclient.get('/application_templates/%s/download' % template_id)
```

クラスタ

この一連の API を使用して、ワークスペース（「アプリケーション」）のメンバーであるクラスタを追加、編集、または削除できます。API キーに関連付けられた `user_role_scope_management` 機能が必要です。

クラスタオブジェクト

クラスタオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	クラスタの一意の識別子。
consistent_uuid	string	自動ポリシー検出実行時の一貫した ID。
application_id	string	クラスタが属するワークスペースの ID。
version	string	クラスタが属するワークスペースのバージョン
name	string	クラスタの名前。
説明	string	クラスタの説明。
approved	boolean	クラスタがユーザーによって「承認」されている場合。
query	JSON	親範囲のフィルタと組み合わせて、フィルタに関連付けられているフィルタ (または一致基準)。

属性	タイプ	説明
short_query{1}JSON{1}	JSON	フィルタに関連付けられているフィルタ(または一致基準)。
alternate_queries	クエリの配列	動的モードで実行される自動ポリシー検出によって生成された代替の推奨クエリ。
インベントリ	インベントリの配列	要求された場合、IP、ホスト名、vrf_id、およびuuidを含むクラスタのメンバーインベントリを返します。

クラスタの取得

このエンドポイントは、特定のワークスペース（「application」）のクラスタのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/clusters
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	クラスタが属するワークスペースの ID。
version	string	クラスタを取得するワークスペースのバージョンの表示。
include_inventory	boolean	クラスタのインベントリを含みます。

応答オブジェクト：この特定のワークスペースとバージョンの全クラスタの配列を返します。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'

restclient.get('/applications/%s/clusters' % application_id)
```

特定のクラスタの取得

このエンドポイントは、クラスタのインスタンスを返します。

```
PUT /openapi/v1/clusters/{cluster_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
cluster_id	string	クラスタの固有識別子。
include_inventory	boolean	クラスタのインベントリを含みます。

応答オブジェクト：指定された ID に関連付けられたクラスタオブジェクトを返します。

サンプル python コード

```
cluster_id = '5d02d021497d4f0949ba74e4'

restclient.get('/clusters/%s' % cluster_id)
```

クラスタの作成

このエンドポイントは、新しいクラスタを作成するために使用されます。

```
POST /openapi/v1/applications/{application_id}/clusters
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	クラスタが属するワークスペースの ID。

JSON クエリの本文には、次のキーが含まれています。

属性	タイプ	説明
name	string	クラスタの名前。
version	string	クラスタが追加されるワークスペースのバージョンを示します。
説明	string	（オプション）クラスタの説明。
approved	boolean	（オプション）承認されたクラスタは、自動ポリシー検出の実行中に更新されません。デフォルトは「False」です。

属性	タイプ	説明
query	JSON	フィルタに関連付けられているフィルタ(または一致基準)。代替クエリモード（動的モードとも呼ばれます）は、ワークスペースで有効にする必要があります。それ以外の場合は無視されます。
query	JSON	フィルタに関連付けられているフィルタ(または一致基準)。代替クエリモード（動的モードとも呼ばれます）は、ワークスペースで有効にする必要があります。それ以外の場合は無視されます。
ノード	配列	IP アドレスまたはエンドポイントのリスト。クエリが提供済みでワークスペースが動的モードでない限り、これらの IP に一致するクエリを作成するために使用されます。

ノード オブジェクトの属性：

名前	タイプ	説明
ip	string	IP アドレス
name	string	（オプション）ノードの名前。
prefix_len	integer	（オプション）サブネットマスク。

（注） クエリが提供済みでワークスペースが動的モードでない限り、ノードはクエリの作成に使用されます。

応答オブジェクト：新しく作成されたクラスタオブジェクトを返します。

サンプル python コード

```
application_id = '5d02b493755f0237a3d6e078'

payload = {
```

```
'name': 'test_cluster',
'version': 'v2',
'description': 'basic granularity',
'approved': False,
'query': {
'type': 'eq',
'field': 'host_name',
'value': 'centos6001'
}
}
restclient.post('/applications/%s/clusters' % application_id)
```

クラスタの更新

このエンドポイントは、クラスタを更新します。

```
PUT /openapi/v1/clusters/{cluster_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
cluster_id	string	クラスタの固有識別子。

JSON クエリの本文には、次のキーが含まれています。

属性	タイプ	説明
name	string	クラスタの名前。
説明	string	（オプション）クラスタの説明。
approved	boolean	承認されたクラスタは、自動ポリシー検出の実行中に更新されません。
query	JSON	フィルタに関連付けられているフィルタ(または一致基準)。代替クエリモード（動的モードとも呼ばれます）は、ワークスペースで有効にする必要があります。それ以外の場合は無視されます。

応答オブジェクト：指定された ID に関連付けられている変更されたクラスタオブジェクトを返します。

サンプル python コード

```
cluster_id = '5d02d2a4497d4f5194f104ef'

payload = {

'name': 'new_test_cluster',

}

restclient.put('/clusters/%s' % cluster_id, json_body=json.dumps(payload))
```

クラスタの削除

このエンドポイントは、指定されたクラスタを削除します。クラスタがポリシーによって使用されている場合、クラスタは削除されず、依存関係のリストが返されます。

```
DELETE /openapi/v1/clusters/{cluster_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
cluster_id	string	クラスタの一意の識別子。

応答オブジェクト：なし

サンプル python コード

```
cluster_id = '5d02d2a4497d4f5194f104ef'

restclient.delete('/clusters/%s' % cluster_id)
```

カンバセーション

カンバセーションは、コンシューマポートが削除された自動ポリシー検出の実行における時間範囲内の集約されたフローです。カンバセーションに関する詳細な説明は、「カンバセーション」を参照してください。

この API を使用すると、特定のワークスペースに対する自動ポリシー検出の実行中に生成されたカンバセーションを検索できます。この API には、API キーに関連付けられている `app_policy_management` 機能が必要です。

ポリシー検出実行での会話の検索

このエンドポイントを使用すると、特定のワークスペースに対する自動ポリシー検出の実行中の会話を検索できます。ダウンロードした会話の一部として表示させる、サポートされているディメンションとメトリックのサブセットを指定することもできます。必要に応じて、サポートされているディメンションとメトリックに関するフィルタを使用して、会話のサブセットに対するクエリを実行できます。

```
POST /openapi/v1/conversations/{application_id}
```

クエリは、次のキーを使用した JSON 本文で構成されます。

名前	タイプ	説明
version	整数	自動ポリシー検出実行のバージョン
filter	JSON	（オプション）クエリフィルタ。フィルタが空（すなわち {}）の場合、クエリはすべての会話に一致します。サポートされているディメンションとメトリックのフィルタを使用して、より具体的な会話をダウンロードできます。フィルタの構文については、フィルタ を参照してください。
寸法	array	（オプション）ダウンロードした会話に対して返されるディメンションのリスト。サポートされているディメンションのリストは、サポートされているディメンション にあります。
メトリック	array	（オプション）ダウンロードした会話に対して返されるメトリックのリスト。サポートされているメトリックのリストは、サポートされているメトリック にあります。
limit	integer	（オプション）1 つの API 応答で返される会話の数。
offset	string	（オプション）前の応答から受信したオフセット（ページネーションに有用）。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
"version": 1,
"filter": {
"type": "and",
"filters":[
```

```
{
"type": "eq",
"field": "excluded",
"value": False
},
{
"type": "eq",
"field": "protocol",
"value": "TCP"
},
]
},
"dimensions": ["src_ip", "dst_ip", "port"],
"metrics": ["byte_count", "packet_count"],
"limit" : 2,
"offset": <offset-object>
}
```

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

キー	値
offset	結果の次のページでパスされる応答オフセット
results	結果のリスト

結果の次のページを生成するには、`offset` の応答で受信したオブジェクトを取得し、それを次のクエリの `offset` の値として渡します。

```
req_payload = {"version": 1,
"limit"：10,
"filter": {"type": "and",
"filters": [
{"type": "eq", "field": "excluded", "value": False},
{"type": "eq", "field": "protocol", "value": "TCP"}
]
}
```

```
}

resp = restclient.post('/conversations/{application_id}', json_body=json.dumps(req_
‹→payload))

print resp.status_code

if resp.status_code == 200:

parsed_resp = json.loads(resp.content)

print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

ポリシー検出実行の上位 N 件の会話

このエンドポイントを使用すると、メトリックに基づき、ディメンションでグループ化された、特定のワークスペースに対して実行された自動ポリシー検出の上位の会話を検索できます。現在サポートされているメトリックはサポートされているメトリック、ディメンションで現在サポートされているグループはサポートされているディメンションです。サポートされているディメンションとメトリックのフィルタを使用して、会話のサブセットをクエリできます。たとえば、バイトトラフィックの会話が最も多いソース IP アドレスを検索するには、`src_ip` ディメンションと `byte_count` メトリックを指定してクエリを使用します。

```
POST /openapi/v1/conversations/{application_id}/topn
```

クエリは、次のキーを使用した JSON 本文で構成されます。

名前	タイプ	説明
version	整数	自動ポリシー検出実行のバージョン
dimension	string	上位 N 個のクエリに対してグループ化される会話のディメンション。 サポートされているディメンション：src_ip、dst_ip
metric	string	上位 N 件の会話に対して並べ替えられるメトリック。サポートされているメトリックのリストは、サポートされているメトリック にあります。

名前	タイプ	説明
filter	JSON	（オプション）クエリフィルタ。フィルタが空（つまり、{}）の場合、クエリはすべての会話に一致します。サポートされているディメンションとメトリックのフィルタを使用して、より具体的な会話をダウンロードできます。フィルタの構文については、フィルタ を参照してください。
しきい値	integer	単一の API 応答で返される上位 N 件の結果の数。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
"version": 1,
"dimension": "src_ip",
"metric": "byte_count",
"filter": {
"type": "and",
"filters":[
{
"type": "eq",
"field": "excluded",
"value": False
},
{
"type": "eq",
"field": "protocol",
"value": "TCP"
},
]
},
"threshold" : 10
}
```

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

キー	値
results	結果キーを持つ 1 つの JSON オブジェクトと、クエリディメンションとメトリックに一致するキーを持つ結果オブジェクトのリストの値をリストします。

```
[ {"result": [
{
"byte_count": 1795195565,
"src_ip": "192.168.1.6"
},
{
"byte_count": 1781002379,
"src_ip": "192.168.1.28"
},
...
] } ]
req_payload = {"version": 1, "dimension": "src_ip", "metric": "byte_count",
"filter": {"type": "and",
"filters": [
{"type": "eq", "field": "excluded", "value": False},
{"type": "eq", "field": "protocol", "value": "TCP"},
{"type": "eq", "field": "consumer_filter_id", "value": "16b12a5614c5af5b68afa7ce
→"},
{"type": "subnet", "field": "src_ip", "value": "192.168.1.0/24"}
]
},
"threshold" : 10
}
resp = restclient.post('/conversations/{application_id}/topn', json_body=json.
→dumps(req_payload))
print resp.status_code
if resp.status_code == 200:
```

```
parsed_resp = json.loads(resp.content)

print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

サポートされているディメンション

名前	タイプ	説明
src_ip	string	コンシューマの IP アドレス
dst_ip	string	コンシューマの IP アドレス
protocol	string	通信で使用されたプロトコル。例：「TCP」、「UDP」など
port	integer	プロバイダーのポート。
address_type	string	「IPv4」または「IPv6」
consumer_filter_id	string	コンシューマ IP がクラスタに属する場合はクラスタのクラスタ ID。それ以外の場合はコンシューマ IP が属する範囲 ID。
provider_filter_id	string	プロバイダー IP がクラスタに属している場合はクラスタのクラスタ ID、それ以外の場合はプロバイダー IP が属する範囲 ID。
excluded	boolean	ポリシーの生成中にこの会話を除外するかどうか。
confidence	double	コンシューマとプロバイダーの分類の信頼度。値は0.0から1.0まで変化し、1.0は分類について高い信頼度を意味します。

サポートされているメトリック

名前	タイプ	説明
byte-count	integer	会話中のバイトの総数
packet_count	integer	会話中のパケットの総数

除外フィルタ

この一連の API は、除外フィルタを追加、編集、または削除するために使用でき、API キーに関連付けられている `user_role_scope_management` 機能が必要です。

除外フィルタは、自動ポリシー検出クラスタリングアルゴリズムからフローを除外します。詳細については、「除外フィルタ」を参照してください。

除外フィルタオブジェクト

除外フィルタオブジェクトの属性について、以下で説明します。

属性	タイプ	説明
id	string	クラスタの一意の識別子。
application_id	string	除外フィルタが属するワークスペースの ID。
version	string	除外フィルタが属するワークスペースのバージョン。
consumer_filter_id	string	定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのコンシューマとして使用できます。
provider_filter_id	string	定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのプロバイダーとして使用できます。
proto	整数	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
port	array	ポートの包含範囲。例：[80, 80] または [5000, 6000]。NULL はすべてのポートを意味します。
updated_at	整数	除外フィルタが更新されたときの UNIX タイムスタンプ。

除外フィルタの取得

このエンドポイントは、特定のワークスペースの除外フィルタのリストを返します。このAPIは、app_policy_management 機能を持つ API キーで使用できます。

```
GET /openapi/v1/applications/{application_id}/exclusion_filters
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。
version	string	除外フィルタを取得するワークスペースのバージョンを示します。

応答オブジェクト：指定されたワークスペースとバージョンの除外フィルタオブジェクトのリストを返します。

サンプル python コード

```
application_id = '<application-id>'

params = {'version': 'v10'}

restclient.get('/applications/%s/exclusion_filters' % application_id,

params=params)
```

特定の除外フィルタを取得する

このエンドポイントは、除外フィルタのインスタンスを返します。

```
GET /openapi/v1/exclusion_filters/{exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
exclude_filter_id	string	除外フィルタの一意の識別子。

応答オブジェクト：指定された ID を持つ除外フィルタオブジェクトを返します。

サンプル python コード

```
exclusion_filter_id = '<exclusion-filter-id>'

restclient.get('/exclusion_filters/%s' % exclusion_filter_id)
```

除外フィルタの作成

このエンドポイントは、除外フィルタを作成するために使用します。

```
POST /openapi/v1/applications/{application_id}/exclusion_filters
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
application_id	string	ワークスペースの一意の識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
version	string	除外フィルタが属するワークスペースのバージョン。
consumer_filter_id	string	（オプション）定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのコンシューマとして使用できます。
provider_filter_id	string	（オプション）定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのプロバイダーとして使用できます。
proto	整数	（オプション）プロトコル整数値（NULL はすべてのプロトコルを意味します）。
start_port	integer	（オプション）範囲の開始ポート。
end_port	integer	（オプション）範囲の終了ポート。

省略可能なパラメータが欠落している場合は、ワイルドカード（いずれにも一致）と見なされます。

応答オブジェクト：作成された除外フィルタオブジェクトを返します。

サンプル python コード

```
provider_filter_id = '<provider-filter-id>'

consumer_filter_id = '<consumer-filter-id>'

payload = {'version': 'v0',

'consumer_filter_id': consumer_filter_id,

'provider_filter_id': provider_filter_id,

'proto': 6,

'start_port': 800,

'end_port': 1000}

application_id = '<application-id>'

restclient.post('/applications/%s/exclusion_filters' % application_id,
```

```
json_body=json.dumps(payload))
```

除外フィルタを更新する

このエンドポイントは除外フィルタを更新します。

```
PUT /openapi/v1/exclusion_filters/{exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
exclude_filter_id	string	除外フィルタの一意の識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
consumer_filter_id	string	（オプション）定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのコンシューマとして使用できます。
provider_filter_id	string	（オプション）定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのプロバイダーとして使用できます。
proto	整数	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
start_port	integer	（オプション）範囲の開始ポート。
end_port	integer	（オプション）範囲の終了ポート。

応答オブジェクト：指定された ID を持つ変更された除外フィルタオブジェクトを返します。

サンプル python コード

```
payload = {'proto': 17}

exclusion_filter_id = '<exclusion-filter-id>'

restclient.post('/exclusion_filters/%s' % exclusion_filter_id,

json_body=json.dumps(payload))
```

除外フィルタの削除

このエンドポイントは、指定された除外フィルタを削除します。

```
DELETE /openapi/v1/exclusion_filters/{exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
exclusion_filter_id	string	除外フィルタの一意の識別子。

応答オブジェクト：なし

サンプル python コード

```
exclusion_filter_id = '<exclusion-filter-id>'

restclient.delete('/exclusion_filters/%s' % exclusion_filter_id)
```

デフォルトの除外フィルタ

この一連の API は、除外フィルタを追加、編集、または削除するために使用できます。API キーに関連付けられている `app_policy_management` 機能が必要です。

除外フィルタは、自動ポリシー検出クラスタリングアルゴリズムからフローを除外します。詳細については、「除外フィルタ」を参照してください。

デフォルト除外フィルタのオブジェクト

除外フィルタオブジェクトの属性について、以下で説明します。

属性	タイプ	説明
id	string	デフォルト除外フィルタの一意の識別子。
consumer_filter_id	string	定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのコンシューマとして使用できます。
provider_filter_id	string	定義されたフィルタの ID。現在、ワークスペース、ユーザー定義フィルタ、または範囲に属するすべてのクラスタをポリシーのプロバイダーとして使用できます。
proto	整数	Protocol Integer value (NULL はすべてのプロトコルを意味します)。

属性	タイプ	説明
port	array	ポートの包含範囲。例：[80, 80] または [5000, 6000]。NULL はすべてのポートを意味します。
updated_at	整数	除外フィルタが更新されたときの UNIX タイムスタンプ。

デフォルトの除外フィルタの取得

このエンドポイントは、デフォルトの除外フィルタのリストを返します。この API は、`app_policy_management` 機能を持つ API キーで使用できます。

```
GET /openapi/v1/default_exclusion_filters?root_app_scope_id={root_app_scope_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_app_scope_id	string	ルート範囲の一意の識別子。

応答オブジェクト：ルート範囲の既定の除外フィルタオブジェクトのリストを返します。

サンプル python コード

特定のデフォルト除外フィルタを取得

このエンドポイントは、デフォルト除外フィルタのインスタンスを返します。

```
default_exclusion_filter_id = '<default-exclusion-filter-id>'

restclient.get('/default_exclusion_filters/%s' % default_exclusion_filter_id)
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
default_exclusion_filter_id	string	除外フィルタの一意の識別子。

応答オブジェクト：指定された ID を持つデフォルトの除外フィルタオブジェクトを返します。

サンプル python コード

```
default_exclusion_filter_id = '<default-exclusion-filter-id>'

restclient.get('/default_exclusion_filters/%s' % default_exclusion_filter_id)
```

デフォルトの除外フィルタの作成

このエンドポイントは、デフォルトの除外フィルタを作成するために使用します。

```
GET /openapi/v1/default_exclusion_filters?root_app_scope_id={root_app_scope_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_app_scope_id	string	ルート範囲の一意の識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
consumer_filter_id	string	（オプション）定義された範囲またはインベントリフィルタの ID。
provider_filter_id	string	（オプション）定義された範囲またはインベントリフィルタの ID。
proto	整数	（オプション）プロトコル整数値（NULL はすべてのプロトコルを意味します）。
start_port	integer	（オプション）範囲の開始ポート。
end_port	integer	（オプション）範囲の終了ポート。

応答オブジェクト：作成されたデフォルトの除外フィルタオブジェクトを返します。

サンプル python コード

```
provider_filter_id = '<provider-filter-id>'
consumer_filter_id = '<consumer-filter-id>'
payload = {'consumer_filter_id': consumer_filter_id,
'provider_filter_id': provider_filter_id,
'proto': 6,
'start_port': 800,
'end_port': 1000}
root_app_scope_id = '<root-app-scope-id>'
restclient.post('/default_exclusion_filters?root_app_scope_id=%s' % root_app_scope_id,
json_body=json.dumps(payload))
```

デフォルトの除外フィルタの更新

このエンドポイントは、デフォルトの除外フィルタを更新します。

```
PUT /openapi/v1/default_exclusion_filters/{default_exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
default_exclusion_filter_id	string	デフォルトの除外フィルタの一意の識別子。

JSON リクエストの本文には、次のキーが含まれています。

属性	タイプ	説明
consumer_filter_id	string	（オプション）定義された範囲またはインベントリフィルタのID。
provider_filter_id	string	（オプション）定義された範囲またはインベントリフィルタのID。
proto	整数	Protocol Integer value (NULL はすべてのプロトコルを意味します)。
start_port	integer	（オプション）範囲の開始ポート。
end_port	integer	（オプション）範囲の終了ポート。

応答オブジェクト：指定された ID を持つ変更されたデフォルトの除外フィルタオブジェクトを返します。

サンプル python コード

```
payload = {'proto': 17}

default_exclusion_filter_id = '<default-exclusion-filter-id>'

restclient.post('/default_exclusion_filters/%s' % default_exclusion_filter_id,

json_body=json.dumps(payload))
```

デフォルトの除外フィルタの削除

このエンドポイントは、指定されたデフォルトの除外フィルタを削除します。

```
DELETE /openapi/v1/default_exclusion_filters/{default_exclusion_filter_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
default_exclusion_filter_id	string	除外フィルタの一意の識別子。

応答オブジェクト：なし

サンプル python コード

```
default_exclusion_filter_id = '<default-exclusion-filter-id>'

restclient.delete('/default_exclusion_filters/%s' % default_exclusion_filter_id)
```

ライブ分析

ライブ分析やポリシー分析は、セキュリティポリシーの生成における重要な側面です。これにより、一連のポリシー（自動ポリシー検出によって生成された場合、またはユーザーによって手動で追加された場合）を評価してから、実際にそれらのポリシーをワークロードに適用できます。ライブ分析を使用すると、アプリケーショントラフィックを中断することなく、ライブトラフィックで what-if 分析を実行できます。

このセクションに記載されている一連のAPIを使用すると、フローをダウンロードし、ワークスペースで現在公開されているの一連のポリシーがそれらのフローに及ぼす影響を確認できます。この一連の API を起動するには、API キーに関連付けられている `app_policy_management` 機能が必要です。

ライブ分析を介して利用できるフローには属性(ディメンションとメトリック)があります。ダウンロードAPIを使用すると、ユーザーはディメンションのさまざまな基準でフローをフィルタリングできます。

ライブ分析でのフローサイズ

このエンドポイントは、ライブ分析を介して利用可能なフローをダウンロードするために、検索条件（またはフィルタ）を指定できる列を把握するために役立ちます。最も一般的なユースケースは、許可されたか、エスケープされたか、または拒否されたフローをダウンロードすることです。カテゴリサイズの検索条件をダウンロードAPIに渡すことでダウンロードが実現します。[type: eq] とともに使用する場合、フローのインバウンドカテゴリおよびアウトバウンドカテゴリが一致している必要があります。[type: contains] とともに使用する場合、フローのインバウンドカテゴリおよびアウトバウンドカテゴリが一致している必要があります。

```
GET /openapi/v1/live_analysis/dimensions
```

ライブ分析で使用可能なフローメトリック

このエンドポイントは、ライブ分析に関連付けられたメトリック（バイト数、パケット数など）のリストを返します。このエンドポイントの 1 つの使用例は、ダウンロード API にメトリックのサブセットを含めることです。つまり、すべてのメトリックをダウンロードする代わりに、ユーザーは関心のあるメトリックの小さなサブセットを指定できます。

```
GET /openapi/v1/live_analysis/metrics
```

ライブ分析を介して利用可能なフローのダウンロード

このエンドポイントは、フィルタ条件に一致するフローのリストを返します。結果に含まれる各フローオブジェクトには、ライブ分析ディメンション（上記のライブ分析ディメンション API によって返される）とライブ分析メトリック（上記のライブ分析メトリック API によって返される）の結合である属性があります。使用可能なディメンションとメトリックの完全なセットに関心がない場合ユーザーは、必要に応じて、ディメンションまたはメトリックの小さなサブセットを指定することもできます。ディメンションまたはメトリックのより小さなサブセットの予測には、API 呼び出しが高速になるという副次的効果もあります。

```
POST /openapi/v1/live_analysis/{application_id}
```

クエリ本文は、次のキーを使用した JSON 本文で構成されます。

名前	タイプ	説明
t0	整数または文字列	時間間隔の開始時刻（エポックまたは ISO 8601）
t1	整数または文字列	時間間隔の終了時刻（エポックまたは ISO 8601）
filter	JSON	クエリ フィルタ。フィルタが空(例: {})の場合、クエリはすべてのフローに一致します。フィルタの構文については、フロー検索の「フィルタ」セクションを参照してください。
寸法	array	（オプション）ライブ分析を介して利用可能なダウンロードされたフローに対して返されるフローディメンションのリスト。指定しない場合、利用可能なすべてのディメンションが返されます。
メトリック	array	（オプション）ライブ分析を介して利用可能なダウンロードされたフローに対して返されるフローメトリックのリスト。
limit	integer	（オプション）単一の API 応答で返される フローの数。
offset	string	（オプション）前の応答から受信したオフセット（ページネーションに有用）。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
"t0": "2016-06-17T09:00:00-0700",
"t1": "2016-06-17T17:00:00-0700",
"filter": {
"type": "and",
```

```
"filters": [
{
"type": "contains",
"field": "category",
"value": "escaped"
},
{
"type": "in",
"field": "dst_port",
"values": ["80", "443"]
}
]
},
"limit": 100,
"offset": <offset-object>
}
```

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

キー	値
offset	結果の次のページでパスされる応答オフセット
results	結果のリスト

結果の次のページを生成するには、`offset` の応答で受信したオブジェクトを取得し、次のクエリの `offset` の値として渡します。

サンプル python コード

```
req_payload = {"t0": "2016-11-07T09:00:00-0700",
"t1": "2016-11-07T19:00:00-0700",
"limit": 10,
"filter": {"type": "and",
"filters": [
{"type": ""contains"," "field": "category", "value": "escaped"},
{"type": "regex", "field": "src_hostname", "value": ""web*"}"
]
```

```
}
}
resp = restclient.post('/live_analysis/{application_id}', json_body=json.dumps(req_
‹→payload))
print resp.status_code
if resp.status_code == 200:
parsed_resp = json.loads(resp.content)
print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

スコープ

この一連の API を使用して、Secure Workload クラスタ展開の範囲（または AppScopes）を管理できます。API キーに関連付けられた `user_role_scope_management` 機能が必要です。範囲リストを取得する API は、 `app_policy_management`または `sensor_management` の機能を持つ API キーで使用できます。

範囲オブジェクト

範囲オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	範囲の固有識別子。
short_name	string	ユーザーが指定した範囲の名前。
name	string	範囲の完全修飾名。これは完全修飾名です。つまり、ルート範囲に至るまで、親範囲の名前 (該当する場合) があります。
説明	string	ユーザーが指定した範囲の説明。
short_query{1}JSON{1}	JSON	範囲に関連付けられているフィルタ (または一致基準)。
query	JSON	（ルート範囲までの）親範囲のフィルタとともに範囲に関連付けられるフィルタ（一致基準）。

属性	タイプ	説明
vrf-id	整数	範囲が属している VRF の ID。
parent_app_scope_id	string	親範囲の ID。
child_app_scope_ids	array	範囲の子 id の配列。
policy_priority		ワークスペースの優先順位を並べ替えるために使用されます。「セマンティクスと表示」を参照してください。
ダーティ	bool	子または親クエリが更新され、変更をコミットする必要があることを示します。
dirty_short_query	JSON	この範囲のクエリが更新されていてもコミットされていない場合は、null 以外です。

範囲の取得

このエンドポイントは、Secure Workload アプライアンスが認識している範囲のリストを返します。この API は、 `app_policy_management` または `user_role_scope_management` のいずれかの機能を持つ API キーで使用できます。

```
GET/openapi/v1/app_scopes
```

パラメータ：

名前	タイプ	説明
vrf-id	整数	アプリの範囲を vrf_id で照合します。
root_app_scope_id	string	アプリの範囲をルートアプリの範囲 ID と照合します。
exact_name	string	大文字と小文字を区別して、正確な名前に一致する範囲を返します。
exact_short_name	string	大文字と小文字を区別して、正確な short_name に一致する範囲を返します。

範囲オブジェクトのリストを戻します。

範囲の作成

このエンドポイントは、新しい範囲を作成するために使用されます。

```
POST/openapi/v1/app_scopes
```

パラメータ：

名前	タイプ	説明
short_name	string	ユーザーが指定した範囲の名前。
説明	string	ユーザーが指定した範囲の説明。
short_query{1}JSON{1}	JSON	範囲に関連付けられているフィルタ (または一致基準)。
parent_app_scope_id	string	親範囲の ID。
policy_priority	integer	デフォルトは「last」です。ワークスペースの優先順位を並べ替えるために使用されます。「ポリシー」の「ポリシーの順序付け」を参照してください。

サンプル python コード

```
req_payload = {

"short_name": "App Scope Name",

"short_query": {

"type":"eq",

"field":"ip",

"value": <....>

},

"parent_app_scope_id": <parent_app_scope_id>

}

resp = restclient.post('/app_scopes', json_body=json.dumps(req_payload))
```

サブネットに基づいて範囲を作成するには、次の short_query を使用します。

```
"short_query":

{

"type":"subnet",
```

```
"field":"ip",
"value": "1.0.0.0/8"
},
```

特定の範囲を取得する

このエンドポイントは、範囲のインスタンスを返します。

`GET/openapi/v1/app_scopes/{app_scope_id}`

指定した ID に関連付けられている範囲オブジェクトを返します。

範囲の更新

このエンドポイントは範囲を更新します。`name` および `description` への変更はすぐに適用されます。short_query への変更は、範囲を「ダーティ」としてマークし、`dirty_short_query` 属性を設定します。特定のルート範囲の下にあるすべての範囲クエリが変更になった場合、範囲クエリの変更のコミットエンドポイントに ping を実行して、必要なすべての更新をコミットする必要があります。

`PUT/openapi/v1/app_scopes/{app_scope_id}`

パラメータ：

名前	タイプ	説明
short_name	string	ユーザーが指定した範囲の名前。
説明	string	ユーザーが指定した範囲の説明。
short_query{1}JSON{1}	JSON	範囲に関連付けられているフィルタ (または一致基準)。

指定された ID に関連付けられている変更された範囲オブジェクトを返します。

特定範囲の削除

このエンドポイントは、指定された範囲を削除します。

`DELETE /openapi/v1/app_scopes/{app_scope_id}`

範囲がワークスペース、ポリシー、ユーザー インベントリ フィルタなどに関連付けられている場合、このエンドポイントは `422 Unprocessable Entity` を返します。返されるエラーオブジェクトには、`details` 属性が含まれ、依存オブジェクトの数とともに各タイプの最初の 10 個の ID が示されます。この情報を使用して、問題となっている依存関係を見つけて削除できます。

ポリシーの優先順位で範囲を取得する

このエンドポイントは、対応するプライマリワークスペースが適用される順序で範囲を一覧表示します。

```
GET /openapi/v1/app_scopes/{root_app_scope_id}/policy_order
```

範囲オブジェクトの配列データを返します。

ポリシー順序の更新

このエンドポイントは、ポリシーが適用される順序を更新します。詳細については、「セマンティクスと表示」を参照してください。

警告 このエンドポイントは、ポリシーが適用される順序を変更します。その結果、新しいホストファイアウォール ルールが挿入され、関連するホスト上で既存のルールがすべて削除されます。

```
POST /openapi/v1/app_scopes/{root_app_scope_id}/policy_order
```

パラメータ：

名前	タイプ	説明
root_app_scope_id	string	順序が変更されるルート範囲。
ids	アレイ	適用される順序での範囲ID文字列の配列。

`ids` 配列パラメータには、ルート範囲のすべてのメンバー（ルートを含む）が含まれている必要があります。

範囲クエリの変更のコミット

このエンドポイントは非同期バックグラウンドジョブをトリガーして、特定のルート範囲のすべての「ダーティ」子を更新します。このジョブは、範囲とワークスペースを更新します。詳細については、「範囲」を参照してください。

```
POST/openapi/v1/app_scopes/commit_dirty
```

パラメータ：

名前	タイプ	説明
root_app_scope_id	string	すべての子が更新されるルート範囲の ID。

名前	タイプ	説明
sync	boolean	（オプション）リクエストを同期する必要があるかどうかを示します。

ジョブがキューに入れられたことを示すために、202を 返します。ジョブが完了したかどうかを確認するには、ルート範囲の「ダーティ」属性をポーリングして、それが false に設定されているかどうかを確認します。

ユーザーは、`sync`パラメータを指定して、ジョブをすぐに実行することもできます。リクエストが完了すると、ステータスコード 200 が返されます。多くの更新を適用する必要がある場合、このリクエストには時間がかかることがあります。

グループ提案リクエストの送信

範囲のグループ提案リクエストを送信します。

```
PUT /openapi/v1/app_scopes/{app_scope_id}/suggest_groups
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
app_scope_id	string	範囲の固有識別子。

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
start_time	string	グループ提案入力時間間隔の開始時間。
end_time	string	グループ提案入力時間間隔の終了時間。

応答オブジェクト：次の属性を持つオブジェクトが返されます。

名前	タイプ	説明
message	string	グループ提案リクエストの提出成否に関するメッセージです。

サンプル python コード

```
app_scope_id = '5d02b493755f0237a3d6e078'

req_payload = {

'start_time': '2020-09-17T10:00:00-0700',

'end_time': '2020-09-17T11:00:00-0700',
```

```
}
resp = restclient.put('/app_scopes/%s/suggest_groups' % app_scope_id,
json_body=json.dumps(req_payload))
```

グループ提案ステータスの取得

範囲のグループ提案ステータスのクエリを実行します。

```
GET /openapi/v1/app_scopes/{app_scope_id}/suggest_groups_status
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
app_scope_id	string	範囲の固有識別子。

応答オブジェクト：次の属性を持つオブジェクトが返されます。

名前	タイプ	説明
status	string	グループ提案のステータス。値：PENDING、COMPLETE、またはFAILED

サンプル python コード

```
app_scope_id = '5d02b493755f0237a3d6e078'
resp = restclient.get('/app_scopes/%s/suggest_groups_status' % app_scope_id)
```

ロール

この一連の API を使用して、ユーザー ロールを管理できます。API キーに関連付けられた `user_role_scope_management` 機能が必要です。

（注） これらの API は、サイト管理者とルート範囲の所有者のみが使用できます。

ロール オブジェクト

ロールオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	ロールの固有識別子。

属性	タイプ	説明
app_scope_id	string	範囲が定義されている範囲。「サービスプロバイダーロール」の場合は空の場合があります。
name	string	ロールのユーザー指定名。
説明	string	ロールのユーザー指定の説明。

ロールの取得

このエンドポイントは、ユーザーがアクセス可能なロールのリストを返します。ロールは、特定のルート範囲にフィルタリングできます。範囲が指定されていない場合、ユーザーがアクセスできるすべての範囲のすべてのロールが返されます。サービス プロバイダ ロールは、ユーザーがサイト管理者である場合にのみ返されます。

```
GET/openapi/v1/roles
```

パラメータ：

名前	タイプ	説明
app_scope_id	string	(オプション) その範囲に割り当てられているロールのみを返すルート範囲の ID。

応答オブジェクト：ユーザーロールオブジェクトのリストを返します。

サンプル python コード

```
resp = restclient.get('/roles')
```

ロールの作成

このエンドポイントは、新しいロールを作成するために使用されます。

```
POST/openapi/v1/roles
```

パラメータ：

名前	タイプ	説明
name	string	ロールのユーザー指定名。
説明	string	ロールのユーザー指定の説明。

名前	タイプ	説明
app_scope_id	string	(オプション) ロールに記載されている範囲 ID がサービス プロバイダ ロールと見なされない場合、ロールが作成される範囲 ID。

要求側のユーザーは、指定された範囲にアクセスできる必要があります。範囲のないロールは「サービス プロバイダ ロール」と呼ばれ、サイト管理者のみが作成できます。

応答オブジェクト：新しく作成されたロールオブジェクトを返します。

サンプル python コード

```
app_scope_id = '<app-scope-id>'

req_payload = {

'name': 'Role Name',

'description': 'Role Description',

'app_scope_id': app_scope_id

}

restclient.post('/roles', json_body=json.dumps(req_payload))
```

特定のロールを取得

このエンドポイントは、特定のロール オブジェクトを返します。

```
GET/openapi/v1/roles/{role_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
role-id	string	ロールの固有識別子。

応答オブジェクト：指定された ID に関連付けられているロールオブジェクトを返します。

サンプル python コード

```
role_id = '<role-id>'

restclient.get('/roles/%s' % role_id)
```

ロールの更新

このエンドポイントは、既存のロールを更新するために使用されます。

```
PUT /openapi/v1/roles/{role_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
role-id	string	ロールの固有識別子。

JSON リクエストの本文には、次のパラメータが含まれています。

名前	タイプ	説明
name	string	ロールのユーザー指定名。
説明	string	ロールのユーザー指定の説明。

要求側のユーザーは、指定された範囲にアクセスできる必要があります。範囲のないロールは「サービスプロバイダーロール」と呼ばれ、サイト管理者のみが更新できます。

応答オブジェクト：指定された ID を持つ更新されたロールオブジェクト。

サンプル python コード

```
role_id = '<role-id>'

req_payload = {

'name': 'Role Name',

'description': 'Role Description',

}

restclient.put('/roles/%s' % role_id, json_body=json.dumps(req_payload))
```

範囲へのロール アクセスを付与

このエンドポイントは、指定されたアクセス レベルを範囲に付与します。

```
POST/openapi/v1/roles/{role_id}/capabilities
```

機能は、ユーザーがアクセスできるロールにのみ追加できます。ロールが範囲に割り当てられている場合、機能はその範囲またはその子に対応している必要があります。サービスプロバイダ ロール (範囲に割り当てられていないもの) は、任意の範囲の機能を追加できます。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
role-id	string	ロールの固有識別子。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
app_scope_id	string	アクセスが提供される範囲の ID。
能力	string	値は次のいずれかです。 SCOPE_READ, SCOPE_WRITE、EXECUTE、ENFORCE、 SCOPE_OWNER、DEVELOPER

機能の詳細については、「ロール」を参照してください。

応答オブジェクト：

名前	タイプ	説明
app_scope_id	string	アクセスが提供される範囲の ID。
role-id	string	ロールの ID。
能力	string	値は次のいずれかです。 SCOPE_READ、 SCOPE_WRITE、EXECUTE、ENFORCE、SCOPE_OWNER、DEVELOPER
継承	boolean	

サンプル python コード

```
role_id = '<role-id>'

req_payload = {

'app_scope_id': '<app-scope-id>',

'ability': 'SCOPE_READ'

}

restclient.post('/roles/%s/capabilities' % role_id,
```

json_body=json.dumps(req_payload))

特定のロールの削除

このエンドポイントは、指定された範囲を削除します。

```
DELETE /openapi/v1/roles/{role_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
role-id	string	ロールの固有識別子。

応答オブジェクト：なし。

サンプル python コード

```
role_id = '<role-id>'

restclient.delete('/roles/%s' % role_id)
```

ユーザ（Users）

この API のセットでユーザーを管理します。API キーに関連付けられた `user_role_scope_management` 機能が必要です。

（注） これらの API は、サイト管理者とルート範囲の所有者のみが使用できます。

ユーザー オブジェクト

ユーザー オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	ユーザー ロールの固有識別子。
電子メール	string	ユーザー アカウントに関連付けられている電子メール。
first_name	string	名前。
last_name	string	姓。
app_scope_id	string	ユーザーに割り当てられる範囲。「サービス プロバイダー ユーザー」の場合、通常は空になります。
role_ids	リスト	ユーザー アカウントに割り当てられているロールの ID のリスト。

属性	タイプ	説明
by-pass_external_auth	boolean	ローカルユーザーの場合は true、外部認証ユーザー（ldap または sso）の場合は false。
disabled_at	整数	ユーザーが無効になっているときの Unix タイムスタンプ。0 または null。そうでない場合。

ユーザーの取得

このエンドポイントは、Secure Workload アプライアンスに認識されているユーザーオブジェクトのリストを返します。

```
GET/openapi/v1/users
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
include_disabled	boolean	（オプション）無効なユーザーを含めるために、デフォルトで false に設定されます。
app_scope_id	string	（オプション）指定した範囲に割り当てられているユーザーのみを返します。

応答オブジェクト：ユーザーオブジェクトのリストを返します。サイト管理者のみが「サービス プロバイダ ユーザー」を表示できます。つまり、範囲に割り当てられていません。

サンプル python コード

```
resp = restclient.get('/users')
```

新しいユーザ アカウントの作成

このエンドポイントは、新しいユーザー アカウントを作成するために使用されます。

```
POST/openapi/v1/users
```

パラメータ：JSON リクエストの本文には、次のパラメータが含まれます。

名前	タイプ	説明
電子メール	string	ユーザー アカウントに関連付けられている電子メール。

名前	タイプ	説明
first_name	string	名前。
last_name	string	姓。
app_scope_id	string	(オプション) ユーザーが属するルート範囲。
role_ids	リスト	(オプション) ユーザーに割り当てられる必要があるロールのリスト。

app_scope_id は、ユーザーが割り当てられるルート範囲の ID です。app_scope_id が表示されない場合、そのユーザーは「サービス プロバイダー ユーザー」です。サイト管理者のみ、サービス プロバイダー ユーザーを作成できます。`role_ids` は、指定されたアプリケーション範囲の下で作成されたロールの ID です。

応答オブジェクト：新しく作成されたユーザーオブジェクトを返します。

サンプル python コード

```
req_payload = {

"first_name": "fname",

"last_name": "lname",

"email": "foo@bar.com"

"app_scope_id": "root_appscope_id",

"role_ids": ["roleid1", "roleid2"]

}

resp = restclient.post('/users', json_body=json.dumps(req_payload))
```

特定のユーザーの取得

このエンドポイントは、特定のユーザー オブジェクトを返します。

```
GET/openapi/v1/users/{user_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	ユーザーオブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられているユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'
```

```
resp = restclient.get('/users/%s' % user_id)
```

ユーザーの更新

このエンドポイントは、既存のユーザーを更新します。

```
PUT /openapi/v1/users/{user_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	更新されるユーザーオブジェクトの ID。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
電子メール	string	ユーザー アカウントに関連付けられている電子メール。
first_name	string	名前。
last_name	string	姓。
app_scope_id	string	ルートアプリケーション範囲ID（サイト管理者にのみ許可）

応答オブジェクト：新しく更新されたユーザーオブジェクトを返します。

サンプル python コード

```
req_payload = {

"first_name": "fname",

"last_name": "lname",

"email": "foo@bar.com"

"app_scope_id": "root_appscope_id",

}

restclient.put('/users', json_body=json.dumps(req_payload))
```

非アクティブ化されたユーザーの有効化/再アクティブ化

このエンドポイントは、非アクティブ化されたユーザーを再び有効化するために使用されます。

```
POST /openapi/v1/users/{user_id}/enable
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	有効化されているユーザーオブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられている、再アクティブ化されたユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'

resp = restclient.post('/users/%s/enable' % user_id)
```

ユーザー アカウントにロールを追加

このエンドポイントは、ユーザー アカウントにロールを追加するために使用されます。

```
PUT/openapi/v1/users/{user_id}/add_role
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	変更されるユーザーオブジェクトの ID。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
role-id	string	追加するロールオブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられている、変更されたユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'

req_payload = {

"role_id": "5ce480d4497d4f1c155d0cef",

}

resp = restclient.put('/users/%s/add_role' % user_id,

json_body=json.dumps(req_payload))
```

ユーザー アカウントからロールを削除

このエンドポイントは、ユーザーアカウントからロールを削除するために使用されます。

/Openapi/v1/users/{user_id}/remove_role の削除

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	削除されるユーザーオブジェクトの ID。

JSON リクエストの本文には、次のパラメータが含まれています

名前	タイプ	説明
role-id	string	削除するロールオブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられている、変更されたユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'

req_payload = {

"role_id": "5ce480d4497d4f1c155d0cef",

}

resp = restclient.delete('/users/%s/remove_role' % user_id,

json_body=json.dumps(req_payload))
```

指定されたユーザーの削除

このエンドポイントは、指定されたユーザー アカウントを削除します。

```
DELETE /openapi/v1/users/{user_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
user_id	string	削除されるユーザーオブジェクトの ID。

応答オブジェクト：指定された ID に関連付けられている、削除されたユーザーオブジェクトを返します。

サンプル python コード

```
user_id = '5ce480db497d4f1ca1fc2b2b'

resp = restclient.delete('/users/%s' % user_id)
```

インベントリ フィルタ

インベントリ フィルタは、インベントリ検索クエリの一致基準をエンコードします。この一連のAPIは、インベントリ フィルタ で説明されているものと同様の機能を提供します。APIキーに関連付けられている `sensor_management or app_policy_management` 機能が必要です。

インベントリ フィルタ オブジェクト

インベントリ フィルタ JSONオブジェクトは、APIエンドポイントに応じて、単一のオブジェクトまたはオブジェクトの配列として返されます。オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	string	インベントリフィルタの一意の識別子。
name	string	ユーザーが指定したインベントリフィルタの名前。
app_scope_id	string	フィルタに関連付けられた範囲の ID。
short_query{1}JSON{1}	JSON	フィルタに関連付けられているフィルタ(または一致基準)。
primary	boolean	「true」の場合は、フィルタが所有権の範囲に制限されていることを意味します。
public	boolean	「true」の場合、フィルタはその範囲にサービスを提供します。また、プライマリ/範囲が制限されている必要があります。

属性	タイプ	説明
query	JSON	親範囲のフィルタと組み合わせて、フィルタに関連付けられているフィルタ (または一致基準)。[所有権の範囲に制限する（restricted to ownership scope）] チェックボックスがオンになっている場合に、これらの連携が有効になります。[primary (プライマリ)] フィールドが false の場合、クエリは short_query と同じです。

インベントリ フィルタの取得

このエンドポイントは、ユーザーに表示されるインベントリ フィルタのリストを返します。

```
GET/openapi/v1/filters/inventories
```

パラメータ：

名前	タイプ	説明
vrf-id	整数	VRF ID でインベントリフィルタを照合します。
root_app_scope_id	string	ルートアプリケーション範囲 ID でインベントリフィルタを照合します。
name	string	大文字と小文字を区別せずに、名前の一部に一致するインベントリフィルタを返します。
exact_name	string	大文字と小文字を区別して、正確な名前に一致するインベントリフィルタを返します。

インベントリ フィルタの作成

このエンドポイントは、インベントリ フィルタを作成するために使用されます。

```
POST/openapi/v1/filters/inventories
```

パラメータ：

名前	タイプ	説明
name	string	アプリケーション範囲のユーザー指定名。
query	JSON	フィルタに関連付けられているフィルタ(または一致基準)。
app_scope_id	string	フィルタに関連付けられたスコープの ID。
プライマリ	boolean	「True」の場合は、フィルタが所有権の範囲に制限されていることを意味します。
public	boolean	「true」の場合、フィルタはそのスコープにサービスを提供します。また、プライマリ/スコープが制限されている必要があります。

サンプル python コード

```
req_payload = {
"app_scope_id": <app_scope_id>,
"name": "sensor_config_inventory_filter",
"query": {
"type": "eq",
"field": "ip",
"value": <sensor_interface_ip>
},
}
resp = restclient.post('/filters/inventories', json_body=json.dumps(req_payload))
```

インベントリフィルタクエリの検証

このエンドポイントは、必要なスキーマに対してクエリの構造を検証します。

```
POST /openapi/v1/filters/inventories/validate_query
```

パラメータ：

名前	タイプ	説明
query	JSON	範囲に関連付けられているフィルタ (または一致基準)。

応答オブジェクト：

属性	タイプ	説明
有効な	boolean	クエリが有効かどうかの表示
errors	アレイ	無効な場合、エラーの詳細

特定のインベントリ フィルタの取得

このエンドポイントは、インベントリフィルタのインスタンスを返します。

```
GET /openapi/v1/filters/inventories/{inventory_filter_id}
```

指定された ID に関連付けられているインベントリ フィルタ オブジェクトを返します。

特定のインベントリフィルタの更新

このエンドポイントは、インベントリフィルタを作成するために使用されます。

```
PUT /openapi/v1/filters/inventories/{inventory_filter_id}
```

パラメータ：

名前	タイプ	説明
name	string	ユーザーが指定した範囲の名前。
query	JSON	範囲に関連付けられているフィルタ (または一致基準)。
app_scope_id	string	フィルタに関連付けられたスコープの ID。
プライマリ	boolean	「True」の場合は、フィルタが所有範囲に制限されていることを意味します。
public	boolean	「true」の場合、フィルタによって特定のサービスが提示されます。ポリシー生成の一部として使用できます。また、プライマリ/範囲が制限されている必要があります。

特定範囲の削除

このエンドポイントは、指定されたインベントリフィルタを削除します。

```
DELETE /openapi/v1/filters/inventories/{inventory_filter_id}
```

フロー検索

フロー検索機能は、「フロー」で説明されているのと同様の機能を提供します。これらの API のセットには、API キーに関連付けられている `flow_inventory_query` 機能が必要です。

フローの寸法のクエリ

このエンドポイントは、フロー検索クエリ (以下) に対して検索条件 (またはフィルタ) を指定できるフローカラムのリストを返します。列の詳細については、「列とフィルタ」を参照してください。

```
GET/openapi/v1/flowsearch/dimensions
```

パラメータ：（なし）

応答オブジェクト：

名前	タイプ	説明
寸法	文字列のリスト	アップロードされたユーザーとオーケストレータ ディメンションのリスト。

サンプル python コード

```
restclient.get('/flowsearch/dimensions')
```

フロー メトリックスのクエリ

このエンドポイントは、フロー観測に関連付けられたメトリックス (バイト数、パケット数など) のリストを返します。

```
GET/openapi/v1/flowsearch/metrics
```

パラメータ：（なし）

応答オブジェクト：

名前	タイプ	説明
メトリック	文字列のリスト	利用可能なメトリックのリスト

サンプル python コード

```
restclient.get('/flowsearch/metrics')
```

フローのクエリ

このエンドポイントは、フィルタ条件に一致するフローのリストを返します。結果の各フローオブジェクトには、フローの大きさ (上記のフロー寸法 API によって返される) とフロー メトリックス (上記のフロー メトリックス API によって返される) の結合である属性があります。

```
POST/openapi/v1/flowsearch
```

フィルタ条件で指定できるカラムのリストは、`/openapi/v1/flowsearch/ dimensions` API を使用して取得できます。

パラメータ：クエリ本文は、次のキーを使用した JSON 本文で構成されます。

名前	タイプ	説明
t0	整数または文字列	フロー検索開始時刻 (エポックまたは ISO 8601)
t1	整数または文字列	フロー検索終了時間 (エポックまたは ISO 8601)
filter	JSON	クエリ フィルタ。フィルタが空 (例: {}) の場合、クエリはすべてのフローに一致します。
scope-name	string	クエリが制限されている範囲の完全な名前。
寸法	array	(オプション) Flowsearch API の結果に返される寸法名のリスト。これは省略可能なパラメータです。指定しない場合、flowsearch の結果は使用可能なすべての寸法を返します。このオプションは、発信者が残りの寸法を気にしない場合に使用可能な寸法のサブセットを指定するときに役立ちます。

名前	タイプ	説明
メトリック	array	(オプション) Flowsearch API の結果で返されるメトリックス名のリスト。これは省略可能なパラメータです。指定しない場合、flowsearch の結果は使用可能なすべてのメトリックスが返されます。このオプションは、発信者が他のメトリックスを処理しない場合に使用可能なメトリックスのサブセットを指定するのに役立ちます。
limit	integer	(オプション) 応答フローの制限数。
offset	string	(オプション) 前の応答から受信した Offset オブジェクト。
降順	boolean	(オプション) このパラメータが false または指定されていない場合、結果はタイムスタンプの昇順になります。パラメータ値が true の場合、結果はタイムスタンプの降順になります。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
"t0": "2016-06-17T09:00:00-0700",
"t1": "2016-06-17T17:00:00-0700",
"filter": {
"type": "and",
"filters": [
"{}",
{
"type": "contains",
"field": "dst_hostname",
"value": "prod"
},
```

```
{
"type": "in",
"field": "dst_port",
"values": ["80", "443"]
}
]
},
"scopeName": "Default:Production:Web",
"limit": 100,
"offset": <offset-object>
}
```

フィルタ

フィルタは、1つ以上のプリミティブ フィルタで構成されるプリミティブ フィルタと論理フィルタ (「not」、「and」、「or」) をサポートします。プリミティブ フィルタの形式は次のとおりです。

```
{"type" : "<OPERATOR>", "field": "<COLUMN_NAME>", "value": "<COLUMN_VALUE>"}
```

プリミティブフィルタの場合は、`eq`、`ne`、`lt`、`lte`、`gt`や`gte`などの比較演算子を使用できます。演算子は、`in`、`regex`、`subnet`、`contains`や`range`にすることもできます。

プリミティブ フィルタの一部の例としては、次のものがあります。

```
{"type": "eq", "field": "src_address", "value": "7.7.7.7"}
{"type": "regex", "field": "src_hostname", "value": ""prod. * "}"
{"type": "subnet", "field": "src_addr", "value": "1.1.11.0/24"}
# 注、' in ' 句は ' value ' ではなく ' values ' キーを使用します
{"type": "in", "field": "src_port", "values": [80, 443]}
```

not、and、orなどのブール演算を使用して、複雑なフィルタを指定することもできます。以下は、これらのタイプのフィルタの一部の例です。

```
# "and" または "or" 演算子は、"filters" のリストを指定する必要があります。
{"type": "and",
"filters": [
{"type": "in", "field": "src_port", "values": [80, 443]},
{"type": "regex", "field": "src_hostname", "value": ""prod. * "}
]"
}
# "not" 演算子は "filter" を指定する必要があります
```

```
{"type": "not",
"filter": {"type": "subnet", "field": "src_addr", "value": "1.1.11.0/24"}
}
```

フロー検索要求での `filter` のスキーマは、より正式に示すと次のようになります。

キー	値
タイプ	フィルタ タイプ
field	プリミティブ フィルタのフィルタ フィールド カラム
filter	フィルタオブジェクト（フィルタタイプ `not` にのみ使用）
filters	フィルタオブジェクトのリスト（フィルタタイプ `and` および `or` に使用）
value	プリミティブ フィルタの値
値	`in` または `range` のフィルタタイプを使用したプリミティブフィルタの値のリスト

プリミティブ フィルタ タイプ

eq、neそれぞれ、「`field`」で指定されたカラム内の「`value`」で指定された値に等価または不等のフローを検索します。次のフィールドをサポートします。`src_hostname`、`dst_hostname`、`src_address`、`dst_address`、`src_port`、`dst_port`、`src_scope_name`、`dst_scope_name`、`vrf_name`、`src_enforcement_epg_name`、`dst_enforcement_epg_name`、`proto` これらの演算子は、ユーザーラベル付きカラムでも機能します。

lt、lte、gt、gte 「`field`」で指定されたカラムの値が「`value`」で指定された値より小さい、値と等しい、値より大きい、値以上である（該当する場合）フローを検索します。次のフィールドをサポートします。`[src_port、dst_port]`

range 「values」リストで指定された範囲の開始と範囲の終了の間で、「field」で指定されたカラムの値のフローを検索します（「values」リストは、「範囲」フィルタータイプのサイズ 2 である必要があります。最初の値は範囲の開始で、2 番目の値は範囲の終了です）。次のフィールドをサポートします。[src_port、dst_port]

in 「`field`」で指定されたカラム内で、「`values`」で指定されたメンバーシップリストに一致するメンバーシップのフローを検索します。次のフィールドをサポートします。`src_hostname`、`dst_hostname`、`src_address`、`dst_address`、`src_port`、`dst_port`、`src_scope_name`、`dst_scope_name`、`vrf_name`、`src_enforcement_epg_name`、`dst_enforcement_epg_name`、`proto` この演算子は、ユーザーラベル付きカラムでも機能します。

regex、containsそれぞれ「`field`」で指定されたカラムで、「`value`」で指定された正規表現の正規表現一致または含む一致のフローを検索します。次のフィールドをサポートします。`src_hostname`、`dst_hostname`、`src_scope_name`、`dst_scope_name`、`vrf_name`、

src_enforcement_epg_name、dst_enforcement_epg_name これらの演算子は、ユーザーラベル付きカラムでも機能します。regex タイプのフィルタは、「value」として Java スタイルの正規表現パターンを使用する必要があります。

subnet CIDR 表記の文字列として「field」で指定されたサブネットメンバーシップのフローを検索します。次のフィールドをサポートしています。[src_address、dst_address]

論理フィルタ タイプ

not 「filter」で指定されたオブジェクトの論理 not フィルタ。

and 「filters」によって指定されたフィルタオブジェクトのリストの論理 and フィルタ

or 「filters」によって指定されたフィルタオブジェクトのリストの論理 or フィルタ

応答オブジェクト：

キー	値
offset	結果の次のページでパスされる応答オフセット
results	結果のリスト

結果の次のページを生成するには、offset の応答で受信したオブジェクトを取得し、それを次のクエリの offset の値として渡します。

サンプル python コード

```
req_payload = {"t0" : "2016-11-07T09:00:00-0700",
"t1" : "2016-11-07T19:00:00-0700",
"scopeName": "Default:Prod:Web",
"limit" : 10,
"filter": {"type": "and",
"filters": [
{"type": "subnet", "field": "src_address", "value": "1.1.11.0/
"‹→24"},
{"type": "regex", "field": "src_hostname", "value": "web * "}
]
}
}
resp = restclient.post('/flowsearch', json_body=json.dumps(req_payload))
print resp.status_code
if resp.status_code == 200:
parsed_resp = json.loads(resp.content)
```

```
print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

フローの TopN クエリ

このエンドポイントは、リスト内のランクが指定されたメトリックスの集約によって決定される場合に、指定された寸法値の上位 n 個のソートされたリストを返します。

```
POST/openapi/v1/flowsearch/topn
```

パラメータ：

フィルタ条件で指定できるカラムのリストは、`/openapi/v1/flowsearch/ dimensions` API を使用して取得できます。要求の本文は、JSON 形式のクエリである必要があります。次に、クエリ本文の例を示します。要求本文内のパラメータ `t0` および `t1` は、エポック形式または iso8601 形式で指定できます。TopN API では、1 日の最大時間範囲のクエリのみが可能です。グループ化を実行する必要があるディメンションは、 `dimension` を使用して指定する必要があります。上位 N 件の結果をランク付けする必要があるメトリックスは、JSON 本文の `metric` フィールドで指定する必要があります。`threshold` には、「上位 N 件」の「N」を示す 1 以上の値を 指定する必要があります。`threshold` の最大値は 1000 です。ユーザーが 1000 を超える値を指定した場合でも、API は最大で 1000 件の結果のみを返します。また、`scopeName` パラメータを指定する必要があります。これは、検索対象を絞り込むための完全な範囲名です。`filter` は、フロー検索（「フィルタ （84 ページ）」）のフィルタと同じです。`filter` を指定しない場合、上位 N 件はすべてのフローエントリが対象になります。

```
{

"t0": "2016-06-17T09:00:00-0700", # t0 を 1466179200 にすることもできます

"t1": "2016-06-17T17:00:00-0700", # t1 を 1466208000 にすることもできます

"dimension": "src_address",

"metric": "fwd_pkts",

"filter": {"type": "eq", "field": "src_address", "value": "172.29.203.193"},

‹→#optional

"threshold": 5,

"scopeName": "Default"

}
```

クエリ本文は、次のキーを使用した JSON 本文で構成されます。

キー	値
t0	フローの開始時刻 (エポックまたは ISO 8601)
t1	フローの終了時刻 (エポックまたは ISO 8601)

キー	値
filter	クエリ フィルタ。フィルタが空(例: {})、またはフィルタが存在しない (オプション) 場合、topN クエリはすべてのフローエントリに適用されます。
scope-name	範囲の完全な名前。クエリの実行対象が指定した範囲のみになります。
dimension	dimension は、グループ化で使用するフィールドです。
metric	metric は dimension の値の合計数です。
threshold	threshold は上位 N 件の「N」の部分です。

応答オブジェクト:

キー	値
result	上位 N エントリの配列

サンプル python コード

```
req_payload = {

"t0": "2017-06-07T08:20:00-07:00",

"t1": "2017-06-07T14:20:00-07:00",

"dimension": "src_address",

"metric": "fwd_pkts",

"filter": {"type": "ne", "field": "src_address", "value": "172.29.203.193"},

"threshold": 5,

"scopeName": "Default"

}

resp = rc.post('/flowsearch/topn',

json_body=json.dumps(req_payload))

print resp.status_code

if resp.status_code == 200:

parsed_resp = json.loads(resp.content)

print json.dumps(parsed_resp)
```

サンプル応答

```
[
```

```
{ "result": [
{"src_address": "172.31.239.163", "fwd_pkts": 23104},
{"src_address": "172.31.239.162", "fwd_pkts": 22410},
{"src_address": "172.31.239.166", "fwd_pkts": 16185},
{"src_address": "172.31.239.168", "fwd_pkts": 15197},
{"src_address": "172.31.239.169", "fwd_pkts": 15116}
]
}
]
```

フローカウント

このエンドポイントは、指定された基準に一致するフロー観測の数を返します。

`POST/openapi/v1/flowsearch/count`

パラメータ：

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。要求本文内のパラメータ `t0` および `t1` は、エポック形式または iso8601 形式で指定できます。この API は、1 日の最大時間範囲のクエリのみを許可します。また、`scopeName` パラメータを指定する必要があります。これは、検索を制限する範囲の完全な名前です。このパラメータが指定されていない場合、フロー観測カウント API 要求は、読み取りアクセス権を持つすべての範囲に適用されます。`filter` は、フロー検索フィルタのフィルタのものと同じです。

```
{
"t0": "2016-06-17T09:00:00-0700", # t0 は 1466179200 にすることもできます
"t1": "2016-06-17T17:00:00-0700", # t1 は 1466208000 にすることもできます
"filter": {"type": "eq", "field": "src_address", "value": "172.29.203.193"},
"scopeName": "Default"
}
```

クエリ本文は、次のキーを使用した JSON 本文で構成されます。

キー	値
t0	フローの開始時刻 (エポックまたは ISO 8601)
t1	フローの終了時刻 (エポックまたは ISO 8601)
filter	クエリ フィルタ。フィルタが空 (例: {}) の場合、クエリはすべてのフローに一致します。
scope-name	クエリが制限されている範囲の完全な名前。

応答オブジェクト：

キー	値
count	フロー検索条件に一致するフロー観測の数。

サンプル python コード

```
req_payload = {
"t0": "2017-07-20T08:20:00-07:00",
"t1": "2017-07-20T10:20:00-07:00",
"scopeName": "Tetration",
"filter": {
"type": "eq",
"field": "dst_port",
"value": "5642"
}
}
resp = rc.post('/flowsearch/count',
json_body=json.dumps(req_payload))
print resp.status_code
if resp.status_code == 200:
parsed_resp = json.loads(resp.content)
print json.dumps(parsed_resp)
```

サンプル応答

```
{"count":508767}
```

インベントリ

インベントリ検索APIは、インベントリ検索で説明されているのと同様の機能を提供します。これらの一連のAPIには、APIキーに関連付けられている `flow_inventory_query` 機能が必要です。

インベントリの寸法のクエリ

このエンドポイントは、インベントリ検索クエリに対して検索条件(またはフィルタ)を指定できるインベントリ カラムのリストを返します。

```
GET/openapi/v1/inventory/search/dimensions
```

インベントリ検索

このエンドポイントは、指定された条件に一致するインベントリ項目のリストを返します。

```
POST/openapi/v1/inventory/search
```

フィルタ条件で指定できるカラムのリストは、/openapi/v1/inventory/ search/dimensions API を使用して取得できます。

パラメータ：

名前	タイプ	説明
filter	JSON	フィルタ クエリ。
scope-name	string	(オプション) 結果を制限する範囲の名前。
limit	integer	（オプション）返される結果の最大数。
offset	integer	(オプション) 次のページを取得する以前の要求からのオフセット。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
"filter": {
"type": "contains",
"field": "hostname",
"value": "collector"
},
"scopeName": "Default:Production:Web", // optional
"limit": 100,
"offset": "<offset-object>" // optional
}
```

サポートされているさまざまなタイプのフィルタを取得するには、フィルタ （84 ページ） を参照してください。

クエリ本文は、次のキーを使用した JSON 本文で構成されます。

キー	値
filter	クエリ フィルタ。フィルタが空 (例: {}) の場合、クエリはすべてのインベントリ項目に一致します。

キー	値
scope-name	クエリが制限される範囲の完全な名前 (オプション)
寸法	インベントリ検索 API の結果で返される寸法名のリスト。これは省略可能なパラメータです。指定しない場合、結果として使用可能なすべての寸法が返されます。このオプションは、発信者が残りの寸法を気にしない場合に使用可能な寸法のサブセットを指定するときに役立ちます。
制限	応答項目の制限数（オプション）
offset	前の応答から受信したオフセットオブジェクト（オプション）

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

名前	タイプ	説明
offset	integer	結果の次のページに渡される応答オフセット。
results	オブジェクトの配列	結果のリスト。

応答には、ページ分割された応答の `offset` フィールドが含まれる場合があります。ユーザーは、次の一連の結果を取得するために、後続の要求で同じオフセットを指定する必要があります。

サンプル Python コード

```
req_payload = {

"scopeName": "Tetration", # optional

"limit": 2,

"filter": {"type": "and",

"filters":[

{"type": "eq", "field": "vrf_name", "value": "Tetration"},

{"type": "subnet", "field": "ip", "value": "1.1.1.0/24"},

{"type": "contains", "field": "hostname", "value": "collector"}

]

}

}
```

```
resp = restclient.post('/inventory/search', json_body=json.dumps(req_payload))
print resp.status_code
if resp.status_code == 200:
parsed_resp = json.loads(resp.content)
print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

インベントリ統計情報

このエンドポイントは、インベントリ項目の統計情報を返します。

```
GET /openapi/v1/inventory/{id}/stats?t0=<t0>&t1=<t1>&td=<td>
```

表 *2:*

パス パラメータ	説明
id	Inventory item id as `{ip}-{vrf_id}` such as `1.1.1.1-123`

クエリー パラメータ	説明
t0	エポック時間の統計情報の開始時刻
t1	エポック時間の統計情報の終了時刻
td	統計情報集約の粒度。整数で秒数を指定します。「分」、「時間」、「日」などの文字列をパスできます。

サンプル python コード

```
resp = restclient.get('/inventory/1.1.1.1-123/stats?t0=1483228800&t1=1485907200&td=day
```

インベントリ カウント

このエンドポイントは、指定された条件に一致するインベントリ項目のリストを返します。

```
POST/openapi/v1/inventory/count
```

フィルタ条件で指定できるカラムのリストは、`/openapi/v1/inventory/ search/dimensions` API を使用して取得できます。

パラメータ：

名前	タイプ	説明
filter	JSON	フィルタ クエリ。

名前	タイプ	説明
scope-name	string	(オプション) 結果を制限する範囲の名前。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
"filter": {
"type": "and",
"filters": [
{
"type": "contains",
"field": "hostname",
"value": "prod"
},
{
"type": "subnet",
"field": "ip"
"value": "6.6.6.0/24"
}
]
},
"scopeName": "Default: Production: Web", # optional
}
```

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

表 *3:*

キー	値
count	フィルタ条件に一致するインベントリ項目の数

サンプル python コード

```
req_payload = {
"scopeName": "Tetration", # optional
"filter": {"type": "and",
```

```
"filters": [
{"type": "eq", "field": "vrf_name", "value": "Tetration"},
{"type": "subnet", "field": "ip", "value": "1.1.1.0/24"},
{"type": "contains", "field": "hostname", "value": "collector"}
]
}
}
resp = restclient.post('/inventory/count', json_body=json.dumps(req_payload))
print resp.status_code
if resp.status_code == 200:
parsed_resp = json.loads(resp.content)
print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

インベントリの脆弱性

このエンドポイントは、脆弱なワークロードに関連付けられた IP アドレスに対応する CVE を返します。

この API は、ルート範囲への最小限の読み取りアクセス権を持つユーザーのみが使用できます。

```
POST /openapi/v1/inventory/cves/{rootScopeID}
```

パラメータ：

名前	タイプ	説明
ips	文字列のリスト	CVE情報を取得するIPのリスト。

要求の本文は、JSON形式のクエリである必要があります。次に、クエリ本文の例を示します。

```
{
"ips": [
"10.18.187.72",
"10.18.187.73"
]
}
```

応答

この応答は、本文の JSON オブジェクトの配列で、次のプロパティが含まれます。

名前	タイプ	説明
ip	string	IP アドレス
cve_ids	文字列のリスト	IP アドレスを持つインベントリの CVE ID のリスト。

サンプル python コード

```
root_scope_id = "5fa0d242497d4f7d968c669b"
req_payload = {
"ips":["10.18.187.72"、"10.18.187.73"]
}
resp = restclient.post('/inventory/cves/' + root_scope_id, json_body=json.dumps(req_
‹→payload))
print resp.status_code
if resp.status_code == 200:
parsed_resp = json.loads(resp.content)
print json.dumps(parsed_resp, indent=4, sort_keys=True)
```

ワークロード

ワークロードAPIは、「ワークロードプロファイル」ページのコンテンツへの、プログラムによるアクセスを提供します。これらの一連の API には、API キーに関連付けられている `sensor_management or flow_inventory_query` 機能が必要です。

ワークロードの詳細

このエンドポイントは、特定のワークロードの指定されたエージェント UUID を返します。

```
GET /openapi/v1/workload/{uuid}
```

パス パラメータ	説明
uuid	エージェント UUID

応答

応答は、指定されたUUIDに関連付けられたワークロードオブジェクトです。ワークロードオブジェクトの属性スキーマを以下に示します。

表 *4 :*

属性	タイプ	説明
agent_type	string	エージェント タイプ
auto_upgrade_opt_out	boolean	true の場合、エージェントはクラスタのアップグレード時に自動的にアップグレードされません。
cpu_quota_mode	integer	CPU クォータ制御
cpu_quota_us	integer	CPU クォータ使用率
current_sw_version	string	ワークロードで実行されているエージェントソフトウェアのバージョン
data_plane_disabled	boolean	true の場合、フローテレメトリデータはエージェントからクラスタにエクスポートされません。
desired_sw_version	string	ワークロードでの実行を意図したエージェントソフトウェアのバージョン
enable_conversation_mode	boolean	true の場合、カンバセーションモードが有効になります。
enable_cache_sidechannel	boolean	true の場合、サイドチャネル攻撃の検出が有効になります。
enable_forensics	boolean	true の場合、フォレンジックが有効になります。
enable_meltdown	boolean	true の場合、メルトダウンエクスプロイト検出が有効になります。
enable_pid_lookup	boolean	true の場合、プロセスルックアップが有効になります。
forensics_cpu_quota_mode	integer	フォレンジック CPU クォータ制御
forensics_cpu_quota_us	integer	フォレンジッククォータの使用状況

属性	タイプ	説明
forensics_mem_quota_bytes	integer	フォレンジック メモリ クォータ（バイト単位）
host_name	string	ワークロードのホスト名
interfaces	アレイ	インターフェイス オブジェクトの配列
kernel_version	string	カーネル バージョン（Kernel version）
last_config_fetch_at	integer	最終設定がフェッチされた日時
last_software_update_at	integer	最終ソフトウェアは、エージェントが現在のバージョンを報告したときのタイムスタンプです。
max_rss_limit	integer	最大メモリ制限
platform	string	ワークロードのプラットフォーム
uuid	string	エージェントの一意の ID
windows_enforcement_mode	string	Windows 強制モードの種類。WAF（Windows Advanced Firewall）または WFP（Windows Filtering Platform）

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'

resp = restclient.get('/workload/%s' % (agent_uuid))
```

ワークロードの統計情報

このエンドポイントは、ワークロードの統計情報を返します。

```
GET /openapi/v1/workload/{uuid}/stats?t0=<t0>&t1=<t1>&td=<td>
```

パス パラメータ	説明
uuid	エージェント UUID

クエリ URL には次のパラメータが含まれています

クエリー パラメータ	説明
t0	エポック時間の統計情報の開始時刻
t1	エポック時間の統計情報の終了時刻終了時刻は開始時刻を 1 日以上超えることができません。
td	統計情報集約の粒度。整数で秒数を指定します。「分」、「時間」、「日」などの文字列をパスできます。

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

名前	タイプ	説明
timestamp	string	メトリックが収集された時刻（エポックまたは ISO 8601）
results	オブジェクト	メトリック

メトリックは、次のプロパティを持つ JSON オブジェクトです

名前	タイプ	説明
flow_count	integer	フロー数。
rx_byte_count	integer	受信バイト数。
rx_packet_count	integer	受信パケット数。
tx_byte_count	integer	送信バイト数。
tx_packet_count	integer	送信パケット数。

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'

td = 15 * 60 # 15 minutes

resp = restclient.get('/workload/%s/stats?t0=1483228800&t1=1485907200&td=%d' % (agent_
‹→uuid, td))

# This code queries workload statistics for a week

t0 = 1483228800

for _ in range(7):

t1 = t0 + 24 * 60 * 60

resp = restclient.get('/workload/%s/stats?t0=%d&t1=%d&td=day' % (agent_uuid, t0,
```

```
,→t1))

t0 = t1
```

インストールされたソフトウェアパッケージ

このエンドポイントは、ワークロードにインストールされているパッケージのリストを返します。

```
GET /openapi/v1/workload/{uuid}/packages
```

パス パラメータ	説明
uuid	エージェント UUID

応答

Thw応答は、パッケージJSONオブジェクトの配列です。パッケージオブジェクトのスキーマは次のとおりです。

属性	タイプ	説明
アーキテクチャ	string	パッケージのアーキテクチャ
name	string	パッケージの名前
パブリッシャ	string	パッケージの発行元
version	string	パッケージのバージョン

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'

resp = restclient.get('/workload/%s/packages' % (agent_uuid))
```

ワークロードの脆弱性

このエンドポイントは、ワークロードで観察された脆弱性のリストを返します。

```
GET /openapi/v1/workload/{uuid}/vulnerabilities
```

脆弱性オブジェクトは、次のキーを使用したJSON本文で構成されます。

パス パラメータ	説明
uuid	エージェント UUID

応答

応答は、脆弱性JSONオブジェクトの配列です。脆弱性オブジェクトのスキーマは次のとおりです。

属性	タイプ	説明
cve_id	string	Common Vulnerability Exposure の ID
package_infos	アレイ	パッケージ情報オブジェクトの配列
v2_score	float	CVSS 2 スコア
v2_access_complexity	string	CVSS V2 アクセスの複雑さ
v2_access_vector	string	CVSS V2 アクセスベクトル
v2_authentication	string	CVSS V2 認証
v2_availability_impact	string	CVSS V2 可用性への影響
v2_confidentiality_impact	string	CVSS V2 機密性への影響
v2_integrity_impact	string	CVSS V2 完全性への影響
v2_severity	string	CVSS V2 重大度
v3_score	float	CVSS V3 スコア
v3_attack_complexity	string	CVSS V3 攻撃の複雑さ
v3_attack_vector	string	CVSS V3 攻撃ベクトル
v3_availability_impact	string	CVSS V3 可用性への影響
v3_base_severity	string	CVSS V3 基本重大度
v3_confidentiality_impact	string	CVSS V2 機密性への影響
v3_integrity_impact	string	CVSS V3 完全性への影響
v3_privileges_required	string	必要な CVSS V3 権限
v3_scope	string	CVSS V3 範囲
v3_user_interaction	string	CVSS V3 ユーザーの操作

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'

resp = restclient.get('/workload/%s/vulnerabilities' % (agent_uuid))
```

ワークロードの長時間実行プロセス

このエンドポイントは、ワークロードで長時間実行されているプロセスのリストを返します。長時間実行プロセスは、稼働時間が 5 分以上のプロセスとして定義されます。

```
GET /openapi/v1/workload/{uuid}/process/list
```

パス パラメータ	説明
uuid	エージェント UUID

応答

応答は、プロセス JSON オブジェクトのリストです。

属性	タイプ	説明
cmd	string	プロセスのコマンド文字列
binary_hash	string	16 進数のプロセスバイナリの SHA256
ctime	long	使用中のプロセスバイナリの ctime
mtime	long	使用中のプロセスバイナリの mtime
exec_path	string	プロセスの実行パス
exit_usec	long	プロセスが使用中に終了した時刻
num_libs	integer	プロセスでロードされるライブラリの数
pid	integer	[プロセス ID（Process ID）]
ppid	integer	親プロセス ID
pkg_info_name	string	プロセスに関連付けられたパッケージの名前
pkg_info_version	string	プロセスに関連付けられたパッケージのバージョン
proc_state	string	プロセスの状態
uptime	long	使用中のプロセスの稼働時間
username	string	プロセスのユーザー名

属性	タイプ	説明
resource_usage	アレイ	リソース使用状況の配列 オブジェクト

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'

resp = restclient.get('/openapi/v1/workload/%s/process/list' % (agent_uuid))
```

ワークロード プロセス スナップショットのサマリー

このエンドポイントは、ワークロード プロセス スナップショットのサマリーを返します。プロセススナップショットには、特定の時間にワークロードによってキャプチャされたすべてのプロセスが含まれています。現在、最新のプロセススナップショットのコピーが1つ保持されています。エンドポイントは、将来の拡張を容易にするために、空のペイロードを持つ POST メソッドをサポートします。

```
POST /openapi/v1/workload/{uuid}/process/tree/ids
```

パス パラメータ	説明
uuid	エージェント UUID

応答

応答は、プロセススナップショットのサマリー JSON オブジェクトのリストです。

属性	タイプ	説明
sensor_uuid	string	エージェント UUID
handle	string	取得するプロセススナップショットへのハンドル
process_count	integer	スナップショットのプロセス数
ts_usec	integer	スナップショットがキャプチャされたときのタイムスタンプ

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'

payload = {

}

resp = restclient.post('/openapi/v1/workload/%s/process/tree/ids' %
```

```
agent_uuid, json_body=json.dumps(payload))
```

ワークロード プロセス スナップショット

このエンドポイントは、ワークロードのプロセススナップショットを返します。プロセススナップショットには、特定の時間にワークロードによってキャプチャされたすべてのプロセスが含まれています。現在、最新のプロセススナップショットのコピーが 1 つ保持されています。このエンドポイントは、ワークロード プロセス スナップショットのサマリーエンドポイントと一緒に使用する必要があります。

```
POST /openapi/v1/workload/{uuid}/process/tree/details
```

パス パラメータ	説明
uuid	エージェント UUID

ペイロードフィールド	タイプ	説明
handle	string	取得するプロセススナップショットへのハンドル

応答

応答は、JSON でのスナップショットに属するプロセスのリストです。

属性	タイプ	説明
command-string	string	トークン化されたコマンド文字列
command_string_raw	string	raw コマンド文字列
binary_hash	string	16 進数のプロセスバイナリの SHA256
ctime	long	使用中のプロセスバイナリの ctime
mtime	long	使用中のプロセスバイナリの mtime
exec_path	string	プロセスの実行パス
process_id	integer	[プロセス ID（Process ID）]
parent_process_id	integer	親プロセス ID
process_key	integer	プロセスの一意のキー
parent_process_key	integer	親プロセスへの一意のキー

属性	タイプ	説明
pkg_info_name	string	プロセスに関連付けられたパッケージの名前
pkg_info_version	string	プロセスに関連付けられたパッケージのバージョン
proc_state	string	プロセスの状態
uptime	long	使用中のプロセスの稼働時間
username	string	プロセスのユーザー名
cve_ids	アレイ	CVEID オブジェクトの配列

サンプル python コード

```
agent_uuid = 'aa28b304f5c79b2f22d87a5af936f4a8fa555894'

payload = {

}

resp = restclient.post('/openapi/v1/workload/%s/process/tree/ids' %
```

agent_uuid, json_body=json.dumps(payload))

```
handle = json.loads(resp.text)['process_summary'][0]['summary'][0]['handle']

payload = {

"handle": handle,

}

resp = restclient.post('/openapi/v1/workload/%s/process/tree/details' %

agent_uuid, json_body=json.dumps(payload))
```

JSON オブジェクトの定義：

インターフェイス

属性	タイプ	説明
ip	string	インターフェイスの IP アドレス
mac	string	インターフェイスの MAC アドレス
name	string	インターフェイスの名前

属性	タイプ	説明
netmask	string	インターフェイスのネットマスク
pcap_opened	boolean	false の場合、インターフェイスのパケットキャプチャは有効化されていません
tags_scope_id	アレイ	インターフェイスに関連付けられた範囲 ID
vrf	string	[VRF名（VRF Name）]
vrf-id	整数	VRF ID

パッケージ情報

属性	タイプ	説明
name	string	パッケージ名
version	string	パッケージのバージョン

リソース使用状況

属性	タイプ	説明
cpu_usage	float	CPU 使用率
memory_usage_kb	integer	メモリ使用量
ts_usec	long	リソース使用量がキャプチャされたときのタイムスタンプ

CVE ID

属性	タイプ	説明
cve_id	string	cve ID
impact__cvss_v2__access_complexity	string	CVE アクセスの複雑度
impact__cvss_v2__access_vector	string	CVE access vector

施行

ポリシーの適用は、生成されたポリシーがワークスペースに関連付けられた範囲内でアセットにプッシュされ、新しいファイアウォールルールが書き込まれる機能です。詳細については、「適用」マニュアルを参照してください。この API のセットには、API キーに関連付けられている `app_policy_management` 機能が必要です。

エージェント ネットワーク ポリシーの設定

このエンドポイントは、エージェント ID に従ってエージェントオブジェクトを返します。これは、ネットワーク ポリシー、エージェント設定、バージョンなどを取得するのに役立ちます。

```
GET /openapi/v1/enforcement/agents/{aid}/network_policy_config
```

パラメータ：

要求 URL には次のパラメータが含まれています。

名前	タイプ	説明
aid	string	ネットワーク ポリシー設定のエージェント UUID。

JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
include_filter_names	boolean	ネットワークポリシーにフィルタ名と ID を含めます。
inject_versions	boolean	ネットワークポリシーに ADM ワークスペースバージョンを含めます。

応答

このエンドポイントの応答は、 エージェント オブジェクトです。

具体的なポリシーの統計情報

このエンドポイントは、エージェント ID と具体的なポリシー ID を指定して、具体的なポリシーの統計情報を返します。エンドポイントは、 Timeseries 具体的ポリシーの結果オブジェクトの配列を返します。

```
GET /openapi/v1/enforcement/agents/{aid}/concrete_policies/{cid}/stats?t0=<t0>&t1=<t1>
↪&td=<td>
```

パラメータ：

リクエスト URL には次のパラメータが含まれています

名前	タイプ	説明
aid	string	統計情報のエージェント UUID
CID	string	統計情報の具体的ポリシー UUID。

JSON クエリの本文には、次のキーが含まれています。

表 5:

名前	タイプ	説明
t0	整数	エポック時間の統計情報の開始時刻
t1	整数	エポック時間の統計情報の終了時刻
td	整数または string	統計情報集約の粒度。整数で秒数を指定します。「分」、「時間」、「日」などの文字列をパスできます。

JSON オブジェクトの定義：

エージェント

属性	タイプ	説明
agent_uuid	string	エージェントの UUID。
agent_config	オブジェクト	エージェント設定
agent_config_status	オブジェクト	エージェント構成ステータス
desired_network_policy_config	object	ネットワーク ポリシー設定
provisioned_network_policy_config	オブジェクト	プロビジョニングされたネットワークポリシー構成
provisioned_state_update_timestamp	integer	エージェントが上記のプロビジョニングされたポリシーを確認したときの秒単位のエポックタイムスタンプ。

属性	タイプ	説明
desired_policy_update_timestamp	integer	desired_network_policy_config が生成されたときの秒単位のエポックタイムスタンプ。
agent_info	オブジェクト	エージェント情報
skipped	boolean	具体的ポリシーの生成がスキップされた場合は true。
message	string	具体的ポリシーの生成がスキップされた理由。

エージェント設定

属性	タイプ	説明
agent_uuid	string	エージェントの UUID。
enforcement_enabled	boolean	エージェントで適用が有効になっていることを示す構成。
fail_mode	string	フェールモード。
version	number	エージェント構成のバージョン番号。
control_tet_rules_only	boolean	tet ルールのみの構成を制御します。
allow_broadcast	boolean	ブロードキャスト構成を許可します。
allow_multicast	boolean	マルチキャスト構成を許可します。
allow_link_local	boolean	リンクローカル構成を許可します。
enforcement_cpu_quota_mode	string	適用エージェントの CPU クォータモード。
enforcement_cpu_quota_us	string	適用エージェントの CPU クォータマイクロ秒。
enforcement_max_rss_limit	number	適用エージェントの最大 RSS 制限。

ネットワーク ポリシー設定

属性	タイプ	説明
version	string	バージョン番号。
network-policy	array	ネットワークポリシー オブジェクトの配列。
address_sets	array	IP セット機能のアドレス セットの配列。
container_network_policy	アレイ	ContainerNetworkPolicy オブジェクトの配列。

ネットワークポリシー

属性	タイプ	説明
priority	string	具体的ポリシーの優先順位。
enforcement_intent_id	string	適用インテント ID。
concrete_policy_id	string	具体的ポリシー ID。
match	オブジェクト	ポリシーの一致基準。これは廃止予定のフィールドです。
action	オブジェクト	ポリシー一致の操作（Action）。
workspace_id	string	ワークスペースの ID。
adm_data_set_id	string	ワークスペースの自動ポリシー検出データセット ID。
adm_data_set_version	string	ワークスペースの自動ポリシー検出データセットバージョン。params に inject_versions=true が渡された場合にのみ設定します。
cluster_edge_id	string	クラスタエッジ ID。
policy_intent_group_id	string	ポリシーインテントグループ ID。

属性	タイプ	説明
match_set	オブジェクト	IP セットサポートの一致セットオブジェクト。一致または match_set の 1 つだけが存在します。
src_filter_id	string	ソースインベントリフィルタ ID。これは、params として include_filter_names=true が渡されたときに設定されます。
src_filter_name	string	ソースインベントリフィルタ名。これは、params として include_filter_names=true が渡されたときに設定されます。
dst_filter_id	string	宛先インベントリフィルタ ID。これは、params として include_filter_names=true が渡されたときに設定されます。
dst_filter_name	string	宛先インベントリフィルタ名。これは、params として include_filter_names=true が渡されたときに設定されます。

ContainerNetworkPolicy

属性	タイプ	説明
pod_id	string	ポッド ID。
network-policy	array	ネットワークポリシー オブジェクトの配列。
deployment	string	展開名。
service_endpoint	アレイ	サービスエンドポイント名のリスト。

一致

属性	タイプ	説明
src_addr	object	サブネットのサブネットオブジェクト。

属性	タイプ	説明
dst-addr	object	サブネットのサブネットオブジェクト。
src_port_range_start	int	送信元ポートの範囲の開始。
src_port_range_end	int	送信元ポート範囲の終了。
dst_port_range_start	int	宛先ポート範囲の開始。
dst_port_range_end	int	宛先ポート範囲の終了。
ip_protocol	string	IP プロトコル。
address-family	string	IPv4 または IPv6 アドレス ファミリ。
direction	string	一致、入力、または出力の方向。
src_addr_range	object	送信元アドレスのアドレス範囲オブジェクト。
dst_add_range	object	宛先アドレスのアドレス範囲オブジェクト。

操作（Action）

属性	タイプ	説明
タイプ	string	アクションタイプ。

一致セット

属性	タイプ	説明
src_set_id	string	ネットワーク ポリシー設定 `address_sets` 配列のアドレスセットオブジェクトの送信元セット ID。 .
dst_set_id	string	ネットワーク ポリシー設定 `address_sets` 配列のアドレスセットオブジェクトの宛先セット ID。

属性	タイプ	説明
src_port	array	送信元ポートのポート範囲オブジェクトの配列。
dst_ports	array	宛先ポートのポート範囲オブジェクトの配列。
ip_protocol	string	IP プロトコル。
address-family	string	IPv4 または IPv6 アドレス ファミリ。
direction	string	一致、入力、または出力の方向。

アドレス セット

属性	タイプ	説明
set-Id	string	アドレスセット ID。
addr_ranges	array	アドレス範囲オブジェクトの配列。
subnets	array	サブネットオブジェクトの配列。
addr_family	string	IPv4 または IPv6 アドレス ファミリ。

サブネット

属性	タイプ	説明
ip_addr	string	[IP Address]。
prefix_length	int	サブネットのプレフィックス長。

アドレス範囲

属性	タイプ	説明
start_ip_addr	string	範囲の開始 IP アドレス。
end_ip_addr	string	範囲の終了 IP アドレス。

ポート範囲

属性	タイプ	説明
start_port	int	範囲の開始ポート。
end_port	int	範囲の終了ポート。

エージェント構成ステータス

属性	タイプ	説明
disabled	boolean	エージェントで適用が無効になっていることを示す構成。
current_version	number	エージェントに適用されているエージェント構成の現在のバージョン。
high_seen_version	number	エージェントが受信したエージェント構成の最新バージョン。

プロビジョニングされたネットワークポリシー構成

属性	タイプ	説明
version	string	エージェントによってプロビジョニングされたネットワークポリシー構成バージョン。
error_reason	string	エージェントがポリシーを正常に適用した場合は CONFIG_SUCCESS、そうでない場合はエラーの理由。
無効	boolean	エージェントで適用が無効になっていることを示す構成。
current_version	number	エージェントに適用されている現在の NPC バージョン。
high_seen_version	number	エージェントが受信した NPC の最高バージョン。
policy_status	オブジェクト	すべてのネットワーク ポリシー ステータス。

エージェント情報

属性	タイプ	説明
agent_info_supported	boolean	エージェント機能で agent_info がサポートされているかどうか。
ipset_supported	boolean	エージェント機能で ipset がサポートされているかどうか。

具体的ポリシーの結果

属性	タイプ	説明
byte-count	int	具体的ポリシーヒットのバイト数。
pkt-count	int	具体的ポリシーヒットのパケット数。

Timeseries 具体的ポリシーの結果

属性	タイプ	説明
timestamp	string	結果集約のタイムスタンプ文字列。
result	object	具体的ポリシーの結果

クライアントサーバー構成

クライアントとサーバーの関係の検出は、Secure Workload のさまざまな機能の中核を成しており、グラウンドトゥルースを報告できるため、可能な場合は常にソフトウェアエージェントを使用することを推奨します。ネットワーク内のテレメトリ モニタリング ポイントでは、さまざまな状況により、特定のフローに関するすべてのパケットの監視は保証されません。たとえば、TCP フローにおける 2 つの単方向の半分がネットワークを通じて一意のパスを通過する可能性があるため、エラーのレベルにより常に影響を受けることを避けられません。

Cisco Secure Workload は、各フローに機械学習アルゴリズムを適用し、一貫性のないテレメトリが報告された場合に判断を提供する統計モデルを構築することにより、ユーザーの操作なしでこれらのエラーを検出して最小限に抑えようとします。ほとんどの場合、ユーザーはこの一連の API について気にする必要はありません。ただし、少数のケースでは、クライアントサーバーの検出アルゴリズムでフローの方向を正しく取得できません。たとえば、自動ポリシー検出など、フローの方向に依存する機能は、不要なポートを開くなどの望ましくない動作を示す場合があります。

既知のサーバーポートに関するヒントを Secure Workload アルゴリズムに提供するために使用できる一連の API が用意されています。この一連の API は、ルート範囲の所有権ロールを持つユーザーが使用できます。それらのユーザーの API キーには、`app_policy_management` 機能が関連付けられている必要があります。

クライアントサーバーの構成には 2 つのオプションがあります。

ホスト構成

ルート範囲内の IP アドレスの特定のサブセットに適用可能な既知のサーバーポートの構成

サーバーポート構成の追加

この API を使用して、特定のルート範囲の既知のサーバーポートに関するヒントを Secure Workload アルゴリズムに提供できます。ルート範囲に属する一連の IP アドレスの既知の TCP/UDP サーバーポートのリストを提供して、フロー内のクライアントサーバーの正しい方向を Secure Workload アルゴリズムに認識させることができます。

```
POST /openapi/v1/adm/{root_scope_id}/server_ports
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲の固有識別子。

さらに、この API への入力として提供されるテキストファイルには、次の形式のエンドポイントサーバーのポート構成が含まれています。

エンドポイントサーバーのポート構成

属性	タイプ	説明
ip_address	string	IP アドレス（IPv4 または IPv6 アドレス）。サブネットは許可されていません。
tcp_server_ports	int のリスト	ip_address に対応する既知の TCP サーバーポートのリスト。
udp_server_ports	int のリスト	ip_address に対応する既知の UDP サーバーポートのリスト。

サーバーポートの一括設定

属性	タイプ	説明
host_config	エンドポイントサーバーのポート構成オブジェクトのリスト。	既知のサーバーポートが関連付けられている IP アドレスのリスト。

サンプル python コード

```
# contents of below file:
# {"host_config": [
# {"ip_address": "1.1.1.1",
# "tcp_server_ports": [100, 101, 102],
# "udp_server_ports": [103]
# },
# {"ip_address": "1.1.1.2",
# "tcp_server_ports": [200, 201, 202]
# }
# ]
# }
file_path = '/<path_to_file>/server_ports.txt'
root_scope_id = '<root-scope-id>'
restclient.upload(file_path,
'/adm/%s/server_ports' % root_scope_id,
timeout=200) # seconds
```

（注） 上記の API は、バックエンドの既知のサーバーポート構成の完全な状態を上書きします。変更が必要な場合は、変更後に完全な構成を再度アップロードする必要があります。

サーバーポート構成の取得

この API は、ルート範囲のエンドポイントに対してアップロードされた既知のサーバーポートのリストを返します。

```
GET /openapi/v1/adm/{root_scope_id}/server_ports
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲の固有識別子。

応答オブジェクト：ref:ServerPortConfig オブジェクトのリスト。

サンプル python コード

```
root_scope_id = '<root-scope-id>'

restclient.get('/adm/%s/server_ports' % root_scope_id)
```

サーバーポート構成の削除

この API は、指定されたルート範囲のサーバーポート構成を削除します。

```
DELETE /openapi/v1/adm/{root_scope_id}/server_ports_config
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_scope_id	string	ルート範囲の固有識別子。

応答オブジェクト：なし。

サンプル python コード

```
"root_scope_id = '<root-scope-id>'

restclient.delete('/adm/%s/server_ports' % root_scope_id)"
```

ポート設定

ルート範囲に属するすべての IP アドレスに適用可能な既知のサーバーポートの設定

サーバーポート設定のプッシュ

このAPIを使用して、特定のルート範囲における既知のサーバーポートに関するヒントをSecure Workloadアルゴリズムに提供できます。ユーザーは、特定のルート範囲における既知のTCP/UDPサーバーポートのリストを提供して、フロー内のクライアントサーバーの正しい方向をSecure Workloadアルゴリズムに把握させることができます。また、各サーバーポートに関連付けられたサービス名を指定するオプションも使用できます。

すべてのルート範囲に適用可能な既知のサービスのデフォルトリストもあります（以降、グローバルサービスと呼びます）。このリストは、ユーザーがいつでも上書きできます。

サービス コンフィギュレーション

サービスは（ポート、名前の）ペアとして定義されます。

属性	タイプ	説明
port	int	TCP/UDPサーバーポート番号
name	string	このポートに関連付けられたサービス名（オプション）

属性	タイプ	説明
override_in_conflicts	boolean	競合が発生した場合にホストを強制的にプロバイダーにする（オプション）

サービスの一括設定

属性	サブ属性	タイプ	説明
server_ports_config	tcp_service_list	サービス設定オブジェクトのリスト	既知の TCP サービスのリスト
	udp_service_list	サービス設定オブジェクトのリスト	既知の UDP サービスのリスト

```
Push services per root scope:

POST /openapi/v1/adm/{root_scope_id}/server_ports_config
```

サンプル python コード

```
# contents of below file: #{"server_ports_config":

#{

#"tcp_service_list": [

#{

#"port": 80,

#"name" : "http"

#

#},

#{

#"port": 53,

#"name" : "dns"

#},

#{

#"port": 514,

#"name" : "syslog",

#"override_in_conflicts": true

#}

#],

#"udp_service_list": [

#{
```

```
#"port": 161
#},
#{
#"port": 53,
#"name" : "dns"
#},
#]
#}
#}
file_path = '/<path_to_file>/server_ports.json'
# Updating service list for a given root scope
#restclient.upload(file_path,
'/openapi/v1/adm/{root_scope_id}/server_ports_config',
timeout=200) # seconds
```

（注） 上記のAPIは、バックエンドの既知のサーバーポート設定の完全な状態を上書きします。ユーザーが何らかの変更を行う場合は、変更後に完全な設定を再度アップロードする必要があります。

サーバーポート設定の取得

このAPIは、ユーザーがアップロードしたルート範囲内における既知のサーバーポートのリストを返します。応答は一括サービス設定です。

```
Retrieve configured services per root scope:
GET /openapi/v1/adm/{root_scope_id}/server_ports_config
Retrieve configured global services:
GET /openapi/v1/adm/server_ports_config
```

サーバーポート設定の削除

このAPIは、指定されたルート範囲のサーバーポート設定を削除します。

```
Remove configured services per root scope:
DELETE /openapi/v1/adm/{root_scope_id}/server_ports_config
```

ソフトウェアエージェント

エージェント API

ソフトウェアエージェント API は、Secure Workload ソフトウェアエージェントの管理に関連付けられています。これらの API のセットには、API キーに関連付けられている `sensor_management` 機能が必要です。以下の GET API は、API キーに関連付けられた `flow_inventory_query` 機能でも使用できます。

ソフトウェア エージェントの取得

このエンドポイントは、ソフトウェア エージェントのリストを返します。

```
GET/openapi/v1/sensors
```

パラメータ：

名前	タイプ	説明
limit	integer	返される結果の数を制限します (オプション)
offset	string	オフセットはページネーション要求に使用されます。応答がオフセットを返す場合、後続の要求では同じオフセットを使用して、次のページでより多くの結果を取得する必要があります。（任意）

特定のソフトウェアエージェントの取得

このエンドポイントは、UUID が URI の一部であるエージェントの属性を返します。

```
GET /openapi/v1/sensors/{uuid}
```

ソフトウェア エージェントの削除

このエンドポイントは、UUID を指定してソフトウェア エージェントを廃止するために使用されます。この API は注意して使用する必要があります。エージェントが削除されると、Secure Workload ダッシュボードに表示されなくなり、エージェントがアクティブな場合、Secure Workload ではエージェントからのフローエクスポートは許可されません。

```
DELETE /openapi/v1/sensors/{uuid}
```

インテントを使用したソフトウェア エージェントの設定

このAPIワークフローは、以下に定義されているいくつかのRESTエンドポイントを使用します。

インベントリ フィルタの作成

このエンドポイントは、ユーザーがソフトウェア エージェントを設定する、エージェント ホストに一致する基準を指定するために使用されます。

```
POST/openapi/v1/filters/inventories
```

パラメータ：

名前	タイプ	説明
app_scope_id	string	インベントリ フィルタに割り当てる範囲 ID。
name	string	インベントリ フィルタの名前。
query	json	エージェント ホストのフィルタまたは一致基準。

サンプル python コード

```
# app_scope_id can be retrieved by /app_scopes API

req_payload = {

"app_scope_id": <app_scope_id>,

"name": "sensor_config_inventory_filter",

"query": {

"type": "eq",

"field": "ip",

"value": <sensor_interface_ip>

}

}

resp = restclient.post('/filters/inventories',

json_body=json.dumps(req_payload)))

print resp.status_code

# 返された応答には、作成されたフィルタとその ID が含まれます。
```

ソフトウェア エージェント設定プロファイルの作成

このエンドポイントは、ソフトウェア エージェントのターゲット セットに適用する一連の設定オプションを指定するために使用されます。

```
POST/openapi/v1/inventory_config/profiles
```

次の設定オプションは、エージェント設定プロファイルの一部として指定できます。

- allow_broadcast：ブロードキャスト トラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- allow_multicast：マルチキャストトラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- allow_link_local：リンクローカルトラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- auto_upgrade_opt_out：True の場合、エージェントは Secure Workload クラスタのアップグレード中に自動アップグレードされません。
- cpu_quota_mode & cpu_quota_usec: これらのオプションは、エンド ホスト上のエージェントに割り当てる CPU クォータの量をポリシングするために使用されます。
- data_plane_disabled：True の場合、エージェントは Cisco Secure Workload へのフローのレポートを停止します。
- enable_conversation_mode：すべてのセンサーでカンバセーションモードを有効にするオプション。
- enable_forensics：ワークロードでのフォレンジックイベントの収集を有効にするオプション（結果的にエージェントはより多くの CPU を使用するようになります）。
- enable_meltdown：ワークロードでの Meltdown エクスプロイト検出を有効にします（結果的にエージェントはより多くの CPU を使用するようになります）。
- enable_pid_lookup: true の場合、エージェントはフローにプロセス情報を接続しようとします。(注) この設定オプションでは、エンド ホストでより多くの CPU が使用されます。
- enforcement_disabled: 適用エージェントを実行しているホストで適用を無効にするために使用できます。
- preserve_existing_rules: 既存の iptable ルールを保持するかどうかを指定するオプション。
- windows_enforcement_mode：WAF（Windows Advanced Firewall）または WFP（Windows Filtering Platform）を使用するオプション（デフォルトのオプションは WAF）。

設定オプションの詳細については、「ソフトウェアエージェント設定」を参照してください。

サンプル python コード

```
# エージェントの data_plane を無効にするためのプロファイルの定義

req_payload = {

"root_app_scope_id": <root_app_scope_id>,
```

```
"data_plane_disabled": True,
"name": "sensor_config_profile_1",
"enable_pid_lookup": True,
"enforcement_disabled": False
}
resp = restclient.post('/inventory_config/profiles',
json_body=json.dumps(req_payload))
print resp.status_code
# 返される応答には、作成されたプロファイルとその ID が含まれます。
parsed_resp = json.loads(resp.content)
```

ソフトウェアエージェント設定プロファイルの取得

このエンドポイントは、ユーザーに表示されるソフトウェアエージェント設定プロファイルのリストを返します。

```
GET /openapi/v1/inventory_config/profiles
```

パラメータ：（なし）

特定のソフトウェアエージェント設定プロファイルの取得

このエンドポイントは、ソフトウェアエージェント設定プロファイルのインスタンスを返します。

```
GET /openapi/v1/inventory_config/profiles/{profile_id}
```

指定した ID に関連付けられているソフトウェアエージェント設定プロファイルオブジェクトを返します。

ソフトウェアエージェント設定プロファイルの更新

このエンドポイントは、ソフトウェアエージェント設定プロファイルを更新します。

```
GET /openapi/v1/inventory_config/profiles/{profile_id}
```

次の設定オプションは、エージェント設定プロファイルの一部として指定できます。

- allow_broadcast：ブロードキャスト トラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- allow_multicast：マルチキャストトラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- allow_link_local：リンクローカルトラフィックを許可/禁止するオプション（このオプションのデフォルト値は True です）。
- auto_upgrade_opt_out：True の場合、エージェントは Secure Workload クラスタのアップグレード中に自動アップグレードされません。

- cpu_quota_mode & cpu_quota_usec: これらのオプションは、エンド ホスト上のエージェントに割り当てる CPU クォータの量をポリシングするために使用されます。
- data_plane_disabled：True の場合、エージェントは Cisco Secure Workload へのフローのレポートを停止します。
- enable_conversation_mode：すべてのセンサーでカンバセーションモードを有効にするオプション。
- enable_forensics：ワークロードでのフォレンジックイベントの収集を有効にするオプション（結果的にエージェントはより多くの CPU を使用するようになります）。
- enable_meltdown：ワークロードでの Meltdown エクスプロイト検出を有効にします（結果的にエージェントはより多くの CPU を使用するようになります）。
- enable_pid_lookup: true の場合、エージェントはフローにプロセス情報を接続しようとします。(注) この設定オプションでは、エンド ホストでより多くの CPU が使用されます。
- enforcement_disabled: 適用エージェントを実行しているホストで適用を無効にするために使用できます。
- preserve_existing_rules: 既存の iptable ルールを保持するかどうかを指定するオプション。
- windows_enforcement_mode：WAF（Windows Advanced Firewall）または WFP（Windows Filtering Platform）を使用するオプション（デフォルトのオプションは WAF）。

設定オプションの詳細については、「ソフトウェアエージェント設定」を参照してください。

指定された ID に関連付けられている、変更されたソフトウェアエージェント設定プロファイルオブジェクトを返します。

ソフトウェアエージェント設定プロファイルの削除

このエンドポイントは、指定されたソフトウェアエージェント設定プロファイルを削除します。

```
DELETE /openapi/v1/inventory_config/profiles/{profile_id}
```

ソフトウェア エージェントの設定インテントの作成

このエンドポイントは、指定された一連のソフトウェアエージェントに一連の設定オプションを適用するインテントを指定するために使用されます。これにより、インテントが作成され、新しく作成されたインテントを順序に追加することによってインテントの順序が更新されます。

```
POST/openapi/v1/inventory_config/intents
```

サンプル python コード

```
req_payload = {

"inventory_config_profile_id": <>,

"inventory_filter_id": <>
```

```
}

resp = restclient.post('/inventory_config/intents',

json_body=json.dumps(req_payload))

print resp.status_code

# 返される応答には、作成されたインテント オブジェクトとその ID が含まれます。
```

インテントの順序の指定

このエンドポイントは、さまざまなソフトウェアエージェントの設定インテントの順序を指定するために使用されます。たとえば、2 つのインテントが存在する場合があります。一方は開発マシンのプロセスIDルックアップを有効にし、他方はWindowsマシンでのプロセスIDルックアップを無効にします。最初のインテントの優先度が高い場合は、開発用の Windows マシンでのプロセスID検索が有効になります。注:デフォルトでは、インテントが作成されると、インテント順序リストの先頭に追加されます。このエンドポイントは、エンドユーザーが既存のインテントの順序を変更する必要がある場合にのみ使用されます。

```
POST/openapi/v1/inventory_config/orders
```

サンプル python コード

```
# エージェント設定インテント順序付きリストを読み取ります。

resp = restclient.get('/inventory_config/orders')

order_result_json = json.loads(resp.content)

# リストに新しいインテントを付加してリストを変更します。

order_rslt_json['intent_ids'].insert(0,<intent_id>)

# 新しい順序をサーバに戻します。

resp = restclient.post('/inventory_config/orders',

json_body=json.dumps(order_rslt_json))
```

エージェント設定インテントの削除

このエンドポイントは、特定のエージェント設定インテントを削除するために使用されます。

```
DELETE /openapi/v1/inventory_config/intents/{intent_id}
```

サンプル python コード

```
intent_id = '588a51dcb5b30d0ee6da084a'

resp = restclient.delete('/inventory_config/intents/%s' % intent_id)
```

インターフェイス構成インテント

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

してインターフェイスに VRF を割り当てることができます。**[管理（Manage）]** > **[エージェント（Agents）]** に移動し、**[設定（Configure）]** タブをクリックします。

Inventory Config Intent オブジェクト

GET および POST メソッドは、inventory config intent JSON オブジェクト配列を返します。オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
vrf-id	整数	VRF ID 整数
vrf_name	string	[VRF名（VRF Name）]
inventory_filter_id	string	インベントリフィルタ ID
inventory_filter	JSON	インベントリフィルタ。詳細については、OpenAPI > インベントリフィルタを参照してください。

インターフェイス構成インテントの取得

このエンドポイントは、ユーザーに表示されるインベントリ設定のリストを返します。

```
GET /openapi/v1/inventory_config/interface_intents
```

パラメータ：（なし）

インターフェイス構成インテントのリストの作成または更新

このエンドポイントは、インターフェイス構成インテントのリストを作成または変更するために使用します。API は、インテントの順序付きリストを受け取ります。このリストからインテントを削除するには、ユーザーは既存のインテントリストを読み込み、変更し、変更したリストを書き戻す必要があります。

```
POST /openapi/v1/inventory_config/interface_intents
```

パラメータ：

名前	タイプ	説明
inventory_filter_id	string	インターフェイスに一致するインベントリフィルタ ID
vrf-id	整数	インターフェイスを割り当てる VRF ID

サンプル python コード

```
req_payload = {

"intents": [
```

```
{"inventory_filter_id": <inventory_filter_id_1>, "vrf_id": <vrf_id_1>},
{"inventory_filter_id": <inventory_filter_id_1>, "vrf_id": <vrf_id_2>}
]
}
resp = restclient.post('/inventory_config/interface_intents', json_body=json.
‹→dumps(req_payload))
```

NAT の背後にあるエージェントの VRF 設定

次の一連の API は、NAT ボックスの背後にあるエージェントに VRF を割り当てるポリシーを指定する場合に役立ちます。これらの API セットには、API キーに関連付けられている `sensor_management` 機能が必要であり、サイト管理ユーザーのみが使用できます。

NAT の背後にあるエージェントの VRF 設定ルールの一覧表示

このエンドポイントは、NAT の背後にあるエージェントに適用可能な VRF 設定ルールのリストを返します。

`GET/openapi/v1/agentnatconfig`

NAT の背後にあるエージェントに適用可能な新しい VRF 設定を作成します

このエンドポイントは、Secure Workload アプライアンスで認識される送信元 IP および送信元ポートに基づいて、ホストの VRF ラベル付け基準を指定するために使用されます。

`POST/openapi/v1/agentnatconfig`

パラメータ：

名前	タイプ	説明
src_subnet	string	送信元 IP が属することができるサブネット (CIDR 表記)。
src_port_range_start	整数	送信元ポート範囲の低バウンド (0-65535)。
src_port_range_end	整数	送信元ポート範囲の高バウンド (0-65535)。
vrf-id	整数	送信元アドレスとポートが上記で指定した範囲内に収まるエージェントのフローのラベル付けに使用される VRF ID。

サンプル python コード

```
req_payload = {
```

```
src_subnet: 10.1.1.0/24, # src IP range for sensors
src_port_range_start: 0,
src_port_range_end: 65535,
vrf_id: 676767 # VRF ID to assign
}
resp = rc.post('/agentnatconfig', json_body=json.dumps(req_payload))
print resp.status_code
```

既存の VRF 設定の削除

```
DELETE /openapi/v1/agentnatconfig/{nat_config_id}
```

Cisco Secure Workload ソフトウェアのダウンロード

Secure Workload ソフトウェアのダウンロード機能では、Secure Workload エージェントのソフトウェアパッケージをダウンロードする方法を提供します。これらの一連の API には、API キーに関連付けられている `software_download` 機能が必要です。この機能は、サイト管理者ユーザー、ルート範囲の所有者、およびエージェントのインストーラロールを持つユーザーのみが使用できます。

サポートされているプラットフォームを取得するための API

このエンドポイントは、サポートされているプラットフォームのリストを返します。

```
GET /openapi/v1/sw_assets/platforms
```

パラメータ：（なし）

応答オブジェクト：サポートされているプラットフォームのリストを返します。

サンプル python コード

以下のサンプルコードは、サポートされているすべてのプラットフォームを取得します。

```
resp = restclient.get('/sw_assets/platforms')
if resp.status_code == 200:
parsed_resp = json.loads(resp.content)
print json.dumps(parsed_resp)
```

サンプル応答

```
{"results": [{"platform": "OracleServer-6.3", "agent_type": "enforcer", "arch": "x86_
‹→64"}, {"platform": "MSWindows8Enterprise", "agent_type": "legacy_sensor", "arch":
‹→"x86_64"}]}
```

サポートされているソフトウェアバージョンを取得するための API

このエンドポイントは、指定された「agent_type」、「package_type」、「platform」、および「architecture」がサポートされているソフトウェアバージョンのリストを返します。

GET /openapi/v1/sw_assets/download?platform=<platform>&agent_type=<agent_type>&pkg_
‹→type=<pkg_type>&arch=<arch>&list_version=<list_version>

ここで <agent_type>、<platform>、<arch> は、**サポートされているプラットフォームを取得するための API** から取得した結果のいずれかにすることができます。<pkg_type> は、「sensor_w_cfg」または「sensor_bin_pkg」のいずれかです。<pkg_type> と <agent_type> はオプションですが、少なくともどちらか 1 つを指定する必要があります。この API を有効にするには、<list_version> が「True」である必要があります。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
platform	string	プラットフォームを指定します。
agent_type	string	（オプション）エージェントの種類を指定します。
pkg_type	string	（オプション）パッケージタイプを指定します。値は「sensor_w_cfg」または「sensor_bin_pkg」のいずれかになります。
arch	string	アーキテクチャを指定します。
list_version	string	ソフトウェアバージョンの検索を有効にするには、「True」に設定します。

応答オブジェクト：サポートされているソフトウェアバージョンのリストを返します。

サンプル python コード

```
resp = restclient.get('/sw_assets/download?platform=OracleServer-6.3&pkg_type=sensor_
‹→w_cfg&arch=x86_64&list_version=True')
if resp.status_code == 200:
print resp.content
```

サンプル応答

```
3.3.1.30.devel
3.3.1.31.devel
```

Secure Workload ソフトウェアをダウンロードするための API

このエンドポイントにより、クライアントは指定された「agent_type」、「package_type」、「platform」、「ar- chitecture」、および「sensor_version」のソフトウェアをダウンロードできます。

```
GET /openapi/v1/sw_assets/download?platform=<platform>&agent_type=<agent_type>&pkg_
‹→type=<pkg_type>&arch=<arch>&sensor_verion=<sensor_version>
```

このとき、<agent_type>、<platform>、<arch> は、**サポート対象のプラットフォームを取得するための API** から得た結果のいずれかにすることができます。<pkg_type> と <agent_type> はオプションですが、少なくともどちらか 1 つを指定する必要があります。<sensor_version> は、**サポート対象のソフトウェアバージョンを取得するための API** から得た結果のいずれかにできます。「sensor_version」が指定されていない場合は、**最新の**ソフトウェアがダウンロードされます。

パラメータ: 要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
platform	string	プラットフォームを指定します。
agent_type	string	（オプション）エージェントの種類を指定します。
pkg_type	string	（オプション）パッケージ タイプを指定します。値は「sensor_w_cfg」または「sensor_bin_pkg」のいずれかになります。
arch	string	アーキテクチャを指定します。
sensor_version	string	（オプション）ソフトウェアのバージョンを指定します。デフォルトは空の文字列です。

応答オブジェクト：指定されたパラメータの Secure Workload ソフトウェアを返します。

サンプル python コード

```
resp = restclient.download('<download_path>/<file_name>','/sw_assets/download?
‹→platform=OracleServer-6.3&pkg_type=sensor_w_cfg&arch=x86_64&sensor_version=3.3.1.30.
‹→devel')
if resp.status_code == 200:
print 'file downloaded successfully'
```

Cisco Secure Workload エージェントのアップグレード

Secure Workload エージェントのアップグレード機能を使用して、インストールされている Secure Workload エージェントを特定のバージョンにアップグレードできます。この機能はメタデータのみを更新し、実際のアップグレードは次のチェックイン時に行われます。API には、API キーに関連付けられた `software_download` 機能が必要です。この機能は、サイト管理者ユーザー、ルート範囲の所有者、またはエージェントのインストーラロールを持つユーザーのみが使用できます。

エージェントを特定のバージョンにアップグレードするための API

このエンドポイントは、「UUID」が特定の「sensor_version」にアップグレードされると、エージェントをトリガーします。「sensor_version」が指定されていない場合は、最新バージョンが適用されます。この API ではダウングレード要求は続行されません。

```
POST /openapi/v1/sensors/{UUID}/upgrade?sensor_version=<sensor_version>
```

<sensor_version> は、「サポートされているソフトウェアバージョンを取得するための API」から取得した結果のいずれかです。

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
sensor_version	string	（オプション）希望のバージョンを指定します。デフォルトでは最新バージョンが適用されます。

このアップグレード要求のステータスを返します。

サンプル python コード

```
resp = restclient.post('/openapi/v1/sensors/{UUID}/upgrade?sensor_version=3.4.1.1.
‹→devel')
if resp.status_code == 200:
print 'agent upgrade was triggered successfully and in progress'
elif resp.status_code == 304:
print 'provided version is not newer than current version'
elif resp.status_code == 400:
print 'provided version is invalid'
elif resp.status_code == 403:
print 'user does not have required capability'
elif resp.status_code == 404:
```

```
print 'agent with {UUID} does not exist'
```

スイッチ

スイッチ関連の API は、Secure Workload ハードウェアエージェントの管理に関連付けられています。これらの API のセットには、API キーに関連付けられている `hw_sensor_management` 機能が必要です。

（注） これらの API は、サイト管理者ユーザーのみが使用できます。

スイッチオブジェクト

スイッチオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
serial	文字列	スイッチのシリアル番号です。
last_checkin_epoch	integer	スイッチを最後に登録したときの UNIX タイムスタンプ。
name	string	スイッチ名です。
ip	string	スイッチの IP アドレス。
nxos_version	string	スイッチの SW バージョン。
agent_version	string	エージェントの SW バージョン。
bootup_time_epoch	integer	スイッチが起動したときの UNIX タイムスタンプ。
ex-port_interval_ms	integer	Secure Workload クラスタへのエクスポート間隔。
datap-ath_disabled	boolean	true の場合、スイッチは Cisco Secure Workload へのフローの報告を停止します
hw_sensors	JSON	`HW` センサーオブジェクトの配列。
catchall_vrf_id	integer	Catch All VRF の ID。

属性	タイプ	説明
role	string	スイッチに関連付けられたロール。
gateway_uuid	string	ゲートウェイ UUID。
deleted_at	整数	スイッチが削除された場合、このパラメータで、オブジェクトが削除されたときのタイムスタンプを得られます。

HW センサー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。

属性	タイプ	説明
name	string	HW センサーの名前。
decommissioned	boolean	デコミッションされた HW センサーの場合は true に設定します。
exporter_id	integer	エクスポータ ID。

スイッチの取得

このエンドポイントは、Secure Workload アプライアンスが認識しているスイッチのリストを返します。

```
GET/openapi/v1/switches
```

パラメータ：なし

応答オブジェクト：スイッチオブジェクトの配列。

サンプル python コード

```
restclient.get('/switches')
```

スイッチの設定

このエンドポイントは、シリアル番号を指定してスイッチを設定するために使用されます。

```
PUT/openapi/v1/switches/{serial}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
serial	文字列	スイッチのシリアル番号です。

クエリ本文は、次のキーを使用したJSON本文で構成されます。キーは、指定したシリアル番号を持つスイッチに対して、次の設定オプションを1つ以上設定するために使用できます。

キー	値
datapath_disabled	オプション パラメータです。trueの場合、スイッチはCisco Secure Workloadへのフローの報告を停止します
export_interval_ms	オプション パラメータです。Secure Workloadクラスタへのエクスポート間隔
catchall_vrf_id	オプション パラメータです。デフォルトのCatch All Vrf Id

応答オブジェクト：なし

サンプル python コード

```
req_payload = {'export_interval_ms': 60000}

resp = restclient.put('/switches/%s' % switch_serial,

json_body=json.dumps(req_payload))
```

スイッチの削除

このエンドポイントは、シリアル番号を指定されたスイッチを削除します。このAPIは慎重に使用する必要があります。

```
DELETE /openapi/v1/switches/{serial}
```

パラメータ：要求URLには、次のパラメータが含まれています。

名前	タイプ	説明
serial	文字列	スイッチのシリアル番号です。

応答オブジェクト：なし

サンプル python コード

```
serial = '<serial>'

restclient.delete('/switches/%s' % serial)
```

収集ルール

これらのAPIのセットを使用して、収集ルールを管理できます。Secure Workloadアプライアンスの収集ルールは、ユーザーが展開でどのようなIPアドレスまたはサブネットを対象とするかを指定するための手段です。展開にSecure Workload分析をサポートするスイッチがある

場合は、これらの収集ルールがスイッチに送信されます（ユーザーは、ダッシュボードの [スイッチに適用（Apply to switches）] チェックボックスをオンにする必要があります）。これらの収集ルールを受信すると、スイッチはこれらの収集ルールセットに一致する IP アドレスのトラフィック信号のみを抽出します。これらの API には、API キーに関連付けられている `hw_sensor_management` 機能が必要です。

（注） これらの API は、サイト管理者ユーザーのみが使用できます。

収集ルール オブジェクト

収集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。

属性	タイプ	説明
subnet	string	CIDR 形式のサブネットまたは IP アドレス。
action	string	有効な値は「INCLUDE」または「EXCLUDE」です。

VRF の新しい収集ルールの更新

このエンドポイントは、指定された VRF の収集ルールの順序付きリストを更新するために使用できます。POST 要求内の収集ルールのリストは、順序付きリストとして扱われる点に注意してください。

```
POST/openapi/v1/collection_rules/{vrf_name}
```

パラメータ：

POST 本文内の収集ルール オブジェクトの順序付きリスト。**最後の 2 つのルールは、IPv4 および IPv6 のすべてのルールをキャッチする必要があります。**ルールでは、次の例のように、サブネット 0.0.0.0/0 と::/0 をそれぞれ指定できます。

応答オブジェクト：VRF の収集ルールの更新済み順序付きリスト。

サンプル python コード

```
req_payload = [

{

"subnet": "10.10.10.0/24",

"action": "INCLUDE"

},

{

"subnet": "11.11.11.0/24",
```

```
"action": "INCLUDE"
},
{
"subnet": "0.0.0.0/0", # catch all rule for IPV4 addresses
"action": "EXCLUDE"
},
{
"subnet": "::/0", # catch all rule for IPV6 addresses
"action": "EXCLUDE"
}
]
resp = restclient.post('/collection_rules/test_vrf', json_body=json.dumps(req_
‹→payload))
```

VRF の収集ルールの取得

このエンドポイントは、指定された VRF の収集ルールの順序付きリストを返します。

```
GET/openapi/v1/collection_rules/{vrf_name}
```

パラメータ：（なし）

応答オブジェクト：指定された VRF の収集ルールの順序付きリスト

サンプル python コード

```
resp = restclient.get('/collection_rules/test_vrf')
```

収集ルールの影響

2 種類のインベントリ項目があります。

- センサー学習（ワークロードプロファイル）：Secure Workload センサーを実行しているワークロードに属するすべての IP アドレスが含まれます。
- フロー学習（インベントリプロファイル）：Secure Workload によって収集されたフロー信号で確認され、Secure Workload エージェントを実行しているいずれのワークロードにも関連付けられていないすべての IP アドレスが含まれます。

EXCLUDE/INCLUDE 収集ルールにより、追跡するインベントリ項目が制御されます。センサー学習インベントリ項目は、収集ルールに関係なく常に追跡されます。収集ルールによって除外されている場合、フロー学習インベントリ項目は存在しません。したがって、インベントリ検索ではそのようなインベントリの結果は返されません。

フロー検索は、収集ルールによって除外された IP に対して入力されないラベル列を除き、収集ルールの影響を受けません。収集ルールは、特定のフローのクライアントとサーバーの判断には影響しません。

収集ルールによって除外された IP のラベルは追跡されないため、自動ポリシー検出の結果に影響が出る可能性があります。

ユーザーがアップロードしたファイルハッシュ

ユーザーはファイルハッシュのリストを Secure Workload にアップロードして、それらのハッシュが良性かフラグ付きかを指定できます。Secure Workload はそれに応じて、それぞれのバイナリハッシュでプロセスにフラグを設定します。

この一連の API を使用して、ファイルハッシュのリストを Cisco Secure Workload にアップロードまたは削除できます。この API を呼び出すには、`user_data_upload` 機能で API キーを使用します。

（注） ルート範囲ごとに最大 100 万のファイルハッシュを持つことができます。良性ハッシュとフラグ付きハッシュについては、それぞれ 500000 のファイルハッシュを持つことができます。

次の API は、範囲所有者とサイト管理者が、「製品」アプライアンスの単一ルート範囲でファイルハッシュをアップロード、ダウンロード、および削除するために使用できます。

ユーザーファイルハッシュのアップロード

このエンドポイントは、Secure Workload アプライアンスのルート範囲のファイルハッシュを持つ CSV ファイルをアップロードする際に使用されます。列ヘッダーの `HashType` と `FileHash` が CSV ファイルに表示されている必要があります。`HashType` は `SHA-1` または `SHA-256` である必要があります。FileHash は空ではなく、40 文字で 16 進数の SHA1 形式または 64 文字で 16 進数の SHA256 形式である必要があります。

`FileName` および `Notes` ヘッダーはオプションです。ファイル名は 150 文字以下、注記は 1024 文字以下にする必要があります。

```
POST /openapi/v1/assets/user_filehash/upload/{rootAppScopeNameOrID}/{benignOrflagged}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeNameOrID	string	ルート範囲の名前または ID。
benignOrflagged	string	`benign` と `flagged` のどちらか 1 つ。

応答オブジェクト：なし

サンプル python コード

```
# Sample CSV File
# HashType,FileHash,FileName,Notes
# SHA-1,1AF17E73721DBE0C40011B82ED4BB1A7DBE3CE29,application_1.exe,Sample Notes
# SHA-256,8F434346648F6B96DF89DDA901C5176B10A6D83961DD3C1AC88B59B2DC327AA4,
→application_2.exe,Sample Notes
file_path = '/<path_to_file>/user_filehash.csv'
root_app_scope_name = 'Tetration'
restclient.upload(file_path, '/assets/user_filehash/upload/%s/benign' % root_app_
→scope_name)
```

ユーザーファイルハッシュの削除

このエンドポイントは、CSV ファイルをアップロードして、Secure Workload アプライアンスのルート範囲からファイルハッシュを削除するために使用されます。CSV ファイルには、ヘッダーとして `FileHash` が必要です。

```
POST /openapi/v1/assets/user_filehash/delete/{rootAppScopeNameOrID}/{benignOrflagged}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeNameOrID	string	ルート範囲の名前または ID。
benignOrflagged	string	`benign` と `flagged` のどちらか 1 つ。

応答オブジェクト：なし

サンプル python コード

```
# Sample CSV File
# FileHash
# 1AF17E73721DBE0C40011B82ED4BB1A7DBE3CE29
# 8F434346648F6B96DF89DDA901C5176B10A6D83961DD3C1AC88B59B2DC327AA4
file_path = '/<path_to_file>/user_filehash.csv'
root_app_scope_name = 'Tetration'
restclient.upload(file_path, '/assets/user_filehash/delete/' + root_app_scope_name +
→'/benign')
```

ユーザーファイルハッシュのダウンロード

このエンドポイントは、Secure Workload アプライアンスの指定されたルート範囲のユーザーファイルハッシュを CSV ファイルとして返します。CSV ファイルには、ヘッダー `HashType`、`FileHash`、`FileName`、および `Notes` がそれぞれの順序で含まれます。

```
GET /openapi/v1/assets/user_filehash/download/{rootAppScopeNameOrID}/{benignOrflagged}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeNameOrID	string	ルート範囲の名前または ID。
benignOrflagged	string	良性またはフラグ付きのいずれかです。

応答オブジェクト：なし

サンプル python コード

```
file_path = '/<path_to_file>/output_user_filehash.csv'

root_app_scope_name = 'Tetration'

restclient.download(file_path, '/assets/user_filehash/download/%s/benign' % root_app_
‹→scope_name)
```

ユーザー定義ラベル

この API は、Secure Workload アプライアンスのフローとインベントリ項目にラベル付けをするためのユーザー定義ラベルを追加または削除するために使用します。この API を呼び出すには、`user_data_upload` 機能で API キーを使用します。フローとインベントリ項目のラベル付けに使用するるキーと値の管理に関するガイドラインについては、UI ユーザーガイドの「ラベルスキーマ」セクションを参照してください。

（注） この機能に UI を使用してアクセスする方法については、「カスタムラベルのインポート」を参照してください。

（注） アップロードできる IPv4/IPv6 アドレスやサブネットの数に関する制限事項については、「ラベルの制限」を参照してください。

範囲依存 API

次の API は、Secure Workload アプライアンスの単一のルート範囲でラベルを取得/設定/削除するために使用されます。これらは、ルート**範囲の所有者**と**サイト管理者**のみが使用できます。さらに、ルート範囲への**読み取りアクセス権**を持つユーザーは、GET API呼び出しを使用できます。

インベントリラベルの取得

このエンドポイントは、Secure Workload アプライアンスのルート範囲にある IPv4/IPv6 アドレスまたはサブネットのラベルを返します。このエンドポイントのクエリに使用されるアドレス/サブネットは、ラベルのアップロードに使用されるものと完全に一致している必要があります。

```
GET /openapi/v1/inventory/tags/{rootAppScopeName}?ip={IPorSubnet}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。
IPorSubnet	string	IPv4/IPv6 アドレスまたはサブネット。

応答オブジェクト：

名前	タイプ	説明
attributes	JSON	フローとインベントリの各項目のマッチングをラベリングするためのキー/値マップ。

サンプル python コード

```
root_app_scope_name = 'Tetration'

restclient.get('/inventory/tags/%s' % root_app_scope_name, params={'ip': '10.1.1.1/24
‹→'})
```

インベントリラベルの検索

このエンドポイントでは、Secure Workload アプライアンスのルート範囲で IPv4/IPv6 アドレスまたはサブネットのラベルを検索できます。

```
GET /openapi/v1/inventory/tags/{rootAppScopeName}/search?ip={IPorSubnet}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

名前	タイプ	説明
IPorSubnet	string	IPv4/IPv6 アドレスまたはサブネット。

応答オブジェクト：この API は、次の形式でオブジェクトのリストを返します。

名前	タイプ	説明
キー	string	IPv4/IPv6 アドレスまたはサブネット。
updatedAt	integer	ラベルが更新されたときの UNIX タイムスタンプ 。
value	JSON	キーの属性のキー/値マップ。

サンプル python コード

```
root_app_scope_name = 'Tetration Scope'

encoded_root_app_scope_name = urllib.quote(root_app_scope_name, safe='')

restclient.get('/inventory/tags/%s/search' % encoded_root_app_scope_name, params={'ip
‹→': '10.1.1.1/24'})
```

インベントリラベルの設定

このエンドポイントは、Secure Workload アプライアンスのルート範囲でフローとインベントリの各項目にラベルを付けるためのラベルの設定に使用されます。

```
POST /openapi/v1/inventory/tags/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
ip	string	IPv4/IPv6 アドレスまたはサブネット。
attributes	JSON	フローとインベントリの各項目のマッチングをラベリングするためのキー/値マップ。

応答オブジェクト：

名前	タイプ	説明
警告	JSON	ラベルの設定中に発生した警告を含むキー/値マップ。

サンプル python コード

```
root_app_scope_name = 'Tetration'

req_payload = {'ip': '10.1.1.1/24', 'attributes': {'datacenter': 'SJC', 'location':
‹→'CA'}}

restclient.post('/inventory/tags/%s' % root_app_scope_name, json_body=json.dumps(req_
‹→payload))
```

インベントリラベルの削除

このエンドポイントは、Secure Workload アプライアンスのルート範囲にある IPv4/IPv6 アドレスまたはサブネットのラベルを削除します。

```
DELETE /openapi/v1/inventory/tags/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
ip	string	IPv4/IPv6 アドレスまたはサブネット。

サンプル python コード

```
root_app_scope_name = 'Tetration'

req_payload = {'ip': '10.1.1.1/24'}

restclient.delete('/inventory/tags/%s' % root_app_scope_name, json_body=json.
‹→dumps(req_payload))
```

ラベルのアップロード

このエンドポイントは、Secure Workload アプライアンスのルート範囲でフローとインベントリの各項目にラベルを設定するためのラベル付きの CSV ファイルをアップロードするために使用されます。`IP` という名前の列ヘッダーが CSV ファイルに表示されている必要があります。残りの列ヘッダーのうち、最大 32 までを使用して、フローとインベントリ項目の注釈を付け

ることができます。ラベルに英語以外の文字を使用するには、アップロードした CSV ファイルが UTF-8 形式である必要があります。

```
POST /openapi/v1/assets/cmdb/upload/{rootAppScopeName}
```

パラメータ：

ユーザーは、この API へのパラメータとして操作タイプ（`X-Tetration-Oper`）を提供する必要があります。`X-Tetration-Oper` は、次のいずれかとすることができます。

- [追加（add）]：ラベルを新規および既存のアドレス/サブネットに追加します。既存のラベルの代わりに新しいラベルを選択して、競合を解決します。たとえば、データベース内の住所のラベルが *{"foo": "1", "bar": "2"}* で、CSV ファイルに *{"z": "1", "bar": "3"}* が含まれている場合、*add* は、このアドレスのラベルを *{"foo": "1", "z": "1", "bar": "3"}* に設定します。
- [上書き（overwrite）]：新しいアドレス/サブネットのラベルを挿入し、既存のラベルを置き換えます。たとえば、データベース内の住所のラベルが *{"foo": "1", "bar": "2"}* で、CSV ファイルに *{"z": "1", "bar": "3"}* が含まれている場合、*overwrite* は、このアドレスのラベルを *{“z”: “1”, “bar”: “3”}* に設定します。
- [マージ（merge）]：ラベルを既存のアドレス/サブネットにマージします。空の値の代わりに空でない値を選択することで、競合を解決します。たとえば、データベース内のアドレスのラベルが *{"foo": "1", "bar": "2", "qux": "", "corge": "4"}* で、CSV ファイルに *{"z": "1", "bar": "", "qux": "3", "corge": "4-updated"}* が含まれている場合、*merge* は、このアドレスのラベルを *{"foo": "1", "z": "1", "bar": "2", "qux": "3", "corge": "4-updated"}* に設定します。

（注） “bar” の値は “” （空）にリセットされず、既存の値 “bar”=”2” が保持されます。

- [削除（delete）]：アドレス/サブネットのラベルを削除します。

応答オブジェクト：

名前	タイプ	説明
警告	JSON	ラベルの設定中に発生した警告を含むキー/値マップ。

サンプル python コード

```
file_path = '/<path_to_file>/user_annotations.csv'

root_app_scope_name = 'Tetration'

req_payload = [tetpyclient.MultiPartOption(key='X-Tetration-Oper', val='add')]

restclient.upload(file_path, '/assets/cmdb/upload/%s' % root_app_scope_name, req_

‹→payload)
```

ユーザーラベルのダウンロード

このエンドポイントは、Secure Workloadアプライアンスのルート範囲に対してユーザーがアップロードしたラベルを CSV ファイルとして返します。

```
GET /openapi/v1/assets/cmdb/download/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答:

Content-Type: *text/csv*

ユーザーが範囲に対してアップロードしたラベルを含む CSV ファイル。

サンプル python コード

```
file_path = '/<path_to_file>/output.csv'

root_app_scope_name = 'Tetration'

restclient.download(file_path, '/assets/cmdb/download/%s' % root_app_scope_name)
```

列ヘッダーの取得

このエンドポイントは、Secure Workloadアプライアンスのルート範囲の列ヘッダーのリストを返します。

```
GET /openapi/v1/assets/cmdb/attributenames/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答オブジェクト：ラベルに使用できるファセットの配列。

サンプル python コード

```
root_app_scope_name = 'Tetration'

resp = restclient.get('/assets/cmdb/attributenames/%s' % root_app_scope_name)
```

列ヘッダーの削除

このエンドポイントは、Secure Workloadアプライアンスのルート範囲の列ヘッダーを削除します。列ヘッダーを削除すると、ラベル付きファセットのリストから削除され、既存のラベルからも削除されます。

```
DELETE /openapi/v1/assets/cmdb/attributenames/{rootAppScopeName}/{attributeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。
attributeName	string	削除対象の属性。

応答オブジェクト：なし

サンプル python コード

```
root_app_scope_name = 'Tetration'

attribute_name = 'column1'

resp = restclient.delete('/assets/cmdb/attributenames/%s/%s' % (root_app_scope_name,

‹→attribute_name))
```

ラベル付きファセットのリストの取得

このエンドポイントは、Secure Workload アプライアンスのルート範囲のラベル付きファセットのリストを返します。ラベル付きファセットは、その範囲内のフローとインベントリ項目に注釈を付けるために使用される、アップロードされた CSV ファイル内の列ヘッダーのサブセットです。

```
GET /openapi/v1/assets/cmdb/annotations/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答オブジェクト：ルート範囲のラベル付きファセットの配列。

サンプル python コード

```
root_app_scope_name = 'Tetration'

resp = restclient.get('/assets/cmdb/annotations/%s' % root_app_scope_name)
```

ラベル付きファセットのリストの更新

このエンドポイントは、Secure Workload アプライアンスのルート範囲のフローとインベントリ項目に注釈を付けるために使用されるファセットのリストを更新します。

```
PUT /openapi/v1/assets/cmdb/annotations/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答オブジェクト：なし

サンプル python コード

```
# 次のリストは、次の列ヘッダーのサブセットです。

# CSV ファイルのアップロード

req_payload = ['location', 'region', 'detail']

root_app_scope_name = 'Tetration'

restclient.put('/assets/cmdb/annotations/%s' % root_app_scope_name,

json_body=json.dumps(req_payload))
```

ユーザーがアップロードしたラベルの消去

このエンドポイントは、Secure Workloadアプライアンスのルート範囲でフローとインベントリの各項目のラベルを消去します。変更は新しいデータに影響します。古いラベル付きデータは変更されずに残ります。

```
POST /openapi/v1/assets/cmdb/flush/{rootAppScopeName}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
rootAppScopeName	string	ルート範囲名。

応答オブジェクト：なし

サンプル python コード

```
root_app_scope_name = 'Tetration'

restclient.post('/assets/cmdb/flush/%s' % root_app_scope_name)
```

範囲に依存しない API

次の API は、Secure Workload アプライアンス上の複数の範囲にまたがる場合があります。

（注） 範囲非依存および範囲依存の注釈付きファセットの数は、どのルート範囲でも 32 を超えてはなりません。

ラベルのアップロード

このエンドポイントは、Secure Workloadアプライアンスのフローとインベントリの各項目にラベルを設定するためのラベル付きの CSV ファイルをアップロードするために使用されます。IP および VRF という名前の列ヘッダーが CSV ファイルに表示され、VRF がラベルのルート範囲と一致する必要があります。残りの列ヘッダーのうち、最大 32 までを使用して、フローとインベントリ項目の注釈を付けることができます。

```
POST/openapi/v1/assets/cmdb/upload
```

パラメータ：

ユーザーは、実行する操作を指定するために、この API のパラメータとして操作タイプ（X-Tetration-Oper）を提供する必要があります。

応答オブジェクト：

名前	タイプ	説明
警告	JSON	ラベルの設定中に発生した警告を含むキー/値マップ。

サンプル python コード

```
file_path = '/<path_to_file>/user_annotations.csv'

req_payload = [tetpyclient.MultiPartOption(key='X-Tetration-Oper', val='add')]

restclient.upload(file_path, '/assets/cmdb/upload', req_payload)
```

ユーザーラベルのダウンロード

このエンドポイントは、Secure Workload アプライアンスのすべての範囲に対してユーザーがアップロードしたラベルを CSV ファイルとして返します。

```
GET/openapi/v1/assets/cmdb/download
```

パラメータ：（なし）

応答:

Content-Type: *text/csv*

ユーザーが範囲に対してアップロードしたラベルを含む CSV ファイル。

サンプル python コード

```
file_path = '/<path_to_file>/output.csv'

restclient.download(file_path, '/assets/cmdb/download')
```

範囲に依存しないラベル

これらのラベルは、特定のルート範囲に関連付けられておらず、アプライアンスのすべての範囲に適用されます。

インベントリラベルの取得

このエンドポイントは、Secure Workload アプライアンスの IPv4/IPv6 アドレスまたはサブネットの範囲に依存しないラベルを返します。このエンドポイントのクエリに使用されるアドレス/サブネットは、ラベルのアップロードに使用されるものと完全に一致している必要があります。

```
GET /openapi/v1/si_inventory/tags?ip={IPorSubnet}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
IPorSubnet	string	IPv4/IPv6 アドレスまたはサブネット。

応答オブジェクト：

名前	タイプ	説明
attributes	JSON	フローとインベントリの各項目のマッチングをラベリングするためのキー/値マップ。

サンプル python コード

```
restclient.get('/si_inventory/tags', params={'ip': '10.1.1.1/24'})
```

インベントリラベルの検索

このエンドポイントにより、Secure Workload アプライアンスの IPv4/IPv6 アドレスまたはサブネットのラベルを検索できます。

```
GET /openapi/v1/si_inventory/tags/search?ip={IPorSubnet}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
IPorSubnet	string	IPv4/IPv6 アドレスまたはサブネット。

応答オブジェクト：この API は、次の形式でオブジェクトのリストを返します。

名前	タイプ	説明
キー	string	IPv4/IPv6 アドレスまたはサブネット。
updatedAt	integer	ラベルが更新されたときの UNIX タイムスタンプ 。
value	JSON	キーの属性のキー/値マップ。

サンプル python コード

```
restclient.get('/si_inventory/tags/search', params={'ip': '10.1.1.1/24'})
```

インベントリラベルの設定

このエンドポイントは、Secure Workloadアプライアンスのフローとインベントリの各項目にラベルを付けるためのラベルの設定に使用されます。

```
POST /openapi/v1/si_inventory/tags
```

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
ip	string	IPv4/IPv6 アドレスまたはサブネット。
attributes	JSON	フローとインベントリの各項目のマッチングをラベリングするためのキー/値マップ。

応答オブジェクト：

名前	タイプ	説明
警告	JSON	ラベルの設定中に発生した警告を含むキー/値マップ。

サンプル python コード

```
req_payload = {'ip': '10.1.1.1/24', 'attributes': {'datacenter': 'SJC', 'location':
‹→'CA'}}
restclient.post('/si_inventory/tags', json_body=json.dumps(req_payload))
```

インベントリラベルの削除

このエンドポイントは、Secure Workload アプライアンスの IPv4/IPv6 アドレスまたはサブネットのラベルを削除します。

```
DELETE /openapi/v1/si_inventory/tags
```

パラメータ：JSON クエリの本文には、次のキーが含まれています。

名前	タイプ	説明
ip	string	IPv4/IPv6 アドレスまたはサブネット。

サンプル python コード

```
req_payload = {'ip': '10.1.1.1/24'}
restclient.delete('/si_inventory/tags, json_body=json.dumps(req_payload))
```

ラベル付きファセットのリストの取得

このエンドポイントは、Secure Workloadアプライアンス上の範囲に依存しないラベル付きファセットのリストを返します。ラベル付きファセットは、すべての範囲のフローおよびインベントリ項目にラベルを付けるために使用される列ヘッダーのサブセットです。

（注） 要求 URL から範囲名を除外して、注釈付き範囲に依存しないファセットのリストを表示および更新します。

```
GET /openapi/v1/assets/cmdb/annotations
```

応答オブジェクト：範囲に依存しないラベル付きファセットの配列。

サンプル python コード

```
resp = restclient.get('/assets/cmdb/annotations')
```

ラベル付きファセットのリストの更新

このエンドポイントは、Secure Workloadアプライアンス上のフローおよびインベントリ項目に注釈を付けるために使用される、範囲に依存しないファセットのリストを更新します。

```
PUT /openapi/v1/assets/cmdb/annotations
```

応答オブジェクト：なし

サンプル python コード

```
# 次のリストは、次の列ヘッダーのサブセットです。
# CSV ファイルのアップロード
req_payload = ['location', 'region', 'detail']
restclient.put('/assets/cmdb/annotations',
json_body=json.dumps(req_payload))
```

Virtual Routing and Forwarding（VRF）

この API のセットは、VRF を管理します。

（注） これらの API は、サイト管理者のみ使用できます。

VRF オブジェクト

VRF オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
id	int	VRF の固有識別子。
name	string	ユーザーが指定した VRF の名前。
tenant_Id	int	親テナントの ID。
switch_vrfs	文字列のリスト	この Secure Workload VRF にマッピングされるスイッチの vrf 名のリスト。
root_app_scope_id	string	関連付けられているルート範囲の ID。
created_At	整数	VRF が作成されたときの Unix タイムスタンプ。
updated_at	整数	VRF が最後に更新されたときの Unix タイムスタンプ。

VRF の取得

このエンドポイントは、VRF のリストを返します。この API は、`sensor_management`、`flow_inventory_query`、または `hw_sensor_management` 機能を備えた API キーで使用できます。

`GET/openapi/v1/vrfs`

パラメータ：（なし）

応答オブジェクト：VRF オブジェクトのリストを返します。

サンプル python コード

```
resp = restclient.get('/vrfs')
```

VRF を作成します

このエンドポイントは、新しい VRF を作成するために使用されます。関連付けられたルート範囲は、VRF ID に一致するクエリを使用して自動的に作成されます。この API は、`sensor_management` 機能を持つ API キーで使用できます。

`POST/openapi/v1/vrfs`

パラメータ：

名前	タイプ	説明
id	int	（オプション）VRF の固有識別子。指定しない場合、Secure Workload クラスタは新しく作成された VRF の一意の ID を生成します。ベストプラクティスは、呼び出し元が一意の ID を明示的に指定する代わりに、Cisco Secure Workload にこれらの ID を生成させることです。
tenant_Id	int	（オプション）親テナントの ID。
name	string	ユーザーが指定した VRF の名前。
switch_vrfs	文字列のリスト	（オプション）この Secure Workload VRF にマッピングされるスイッチの vrf 名のリスト。
apply_monitoring_rules	boolean	（オプション）収集ルールを VRF に適用するかどうかを指定します。デフォルトは「false」です。詳細については、「収集ルール」を参照してください。

`tenant_id` はオプションです。指定しない場合、VRF は VRF と同じ ID を持つテナント（必要に応じて自動作成される）に追加されます。`tenant_id` が指定されている場合、テナントは自動作成されず、テナントが存在しない場合はエラーが返されます。

応答オブジェクト：新しく作成された VRF オブジェクトを返します。

サンプル python コード

```
req_payload = {

"tenant_id": <tenant_id>,

"name": "Test",

"apply_monitoring_rules": True

}

resp = restclient.post('/vrfs', json_body=json.dumps(req_payload))
```

特定の VRF の取得

このエンドポイントは、指定された VRF ID の情報を返します。この API は、sensor_management、flow_inventory_query、または hw_sensor_management 機能を備えた API キーで使用できます。

```
GET /openapi/v1/vrfs/{vrf_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
vrf-id	int	VRF の固有識別子。

応答オブジェクト：指定された ID に関連付けられている VRF オブジェクトを返します。

サンプル python コード

```
vrf_id = 676767

resp = restclient.get('/vrfs/%d'% vrf_id)
```

VRF の更新

このエンドポイントは、VRF を更新します。この API は、sensor_management 機能を備えた API キーで使用できます。

```
PUT/openapi/v1/vrfs/{vrf_id}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
vrf-id	int	VRF の固有識別子。

JSON リクエストの本文には、次のパラメータが含まれています。

名前	タイプ	説明
name	string	ユーザーが指定した VRF の名前。
switch_vrfs	文字列のリスト	（オプション）この Secure Workload VRF にマッピングされるスイッチの VRF 名のリスト。
apply_monitoring_rules	boolean	（オプション）収集ルールを VRF に適用するかどうかを指定します。

応答オブジェクト：指定された ID に関連付けられている変更された VRF オブジェクトを返します。

サンプル python コード

```
vrf_id = 676767

req_payload = {

"name": "Test",

"apply_monitoring_rules": True

}

resp = restclient.put('/vrfs/%d'% vrf_id,

json_body=json.dumps(req_payload))
```

特定の VRF の削除

このエンドポイントは、VRF を削除します。ルート範囲が関連付けられている場合、これは失敗します。この API は、`sensor_management` 機能を備えた API キーで使用できます。

```
DELETE /openapi/v1/vrfs/{vrf_id}
```

パラメータ：次のパラメータは URL の一部です

名前	タイプ	説明
vrf-id	int	VRF の固有識別子。

サンプル python コード

```
vrf_id = 676767

resp = restclient.delete('/vrfs/%d'% vrf_id)
```

オーケストレーション

この一連の API を使用して、Secure Workload クラスタ展開での外部オーケストレータのインベントリ学習を管理できます。API キーに関連付けられた `external_integration` 機能が必要です。

現在サポートされているオーケストレーションタイプは、「vcenter」（VCenter 6.5 以降）、「kubernetes」、「dns」、「f5」、「netscaler」、「infoblox」および「Cisco FMC」です。外部オーケストレータにあるサポートされるユーザーインターフェイス。

オーケストレーション オブジェクト

オーケストレータ オブジェクトの属性について次に説明します。一部のフィールドは、特定のオーケストレータタイプにのみ適用されます。以下の表に制限事項を示します。

属性	タイプ	説明
id	string	オーケストレーションの固有識別子。
name	string	ユーザーが指定したオーケストレーションの名前。
type	string	オーケストレータのタイプ：サポートされる値（vcenter、kubernetes、f5、netscaler、infoblox、dns）
説明	string	オーケストレーションのユーザー指定の説明。
username	string	オーケストレーション エンドポイントのユーザー名。（dnsの場合は不要）
password	string	オーケストレーション エンドポイントのパスワード。（dnsの場合は不要）
証明書	string	認証に使用されるクライアント証明書（dnsの場合は不要）
キー	string	クライアント証明書に対応するキー（dns の場合は不要）
ca_certificate	string	オーケストレーション エンドポイントを検証する CA 証明書（dns の場合は不要）
auth_token	string	不透明な認証トークン（ベアラートークン）（kubernetesにのみ適用）
insecure	boolean	厳密な SSL 検証を無効にする
delta_interval	整数	秒単位のデルタポーリング間隔の Secure Workload Inventory Managerは、delta_interval秒ごとに増分変更のポーリングを実行します。このパラメータは、Infoblox および Cisco Secure Firewall Management Center には適用されないことに注意してください。

属性	タイプ	説明
full_snapshot_interval	整数	秒単位のフルスナップショット間隔の Secure Workload Inventory Manager は、オーケストレータからフル更新ポーリングを実行します。
verbose_tsdb_metrics	boolean	エンドポイントごとの TSDB メトリックス
hosts_list	Array	オーケストレータへの Secure Workload の接続方法を指定する { “host_name”, port_number} ペアの配列
use_secureconnector_tunnel	boolean	Secure Connector トンネルを介したこのオーケストレータのホストへのトンネル接続
route_domain	integer	F5 ロードバランサでポーリングするルートドメイン番号（「f5」にのみ適用）
dns_zones	配列	DNS サーバーからポーリングする DNS ゾーンを含む文字列の配列（ dns のみ）。各 DNS ゾーンエントリは、a で終わる必要があります。
enable_enforcement	boolean	ファイアウォールやロードバランサなど、ポリシーの適用をサポートする外部オーケストレータにのみ適用されます。たとえば、Cisco Secure Firewall Management Center、F5 BIGIP、および Citrix Netscaler などがあります。デフォルトでは、このフラグは false（ポリシーの強制は無効）です。true の場合、ワークスペースに対してポリシーの適用が実行されると、外部オーケストレータは指定されたロードバランサ アプライアンスにポリシーを展開します。

属性	タイプ	説明
ingress_controllers	オブジェクト	Ingress コントローラ オブジェクトの配列。
fmc_enforcement_mode	string	Cisco Secure Firewall Management Center 外部オーケストレータにのみ適用され、マージ（デフォルト）またはオーバーライドのいずれかである必要があります。最初のインスタンスは、既存のプレフィルタルールより先にすべての Secure Workload ポリシールールを配置するように、Cisco Secure Firewall Management Center ポリシーエンフォーサに指示します。後者のインスタンスは、ユーザーによって作成されたすべてのプレフィルタルールを削除します。
infoblox_config	オブジェクト	Infoblox外部オーケストレータにのみ適用されます。Infoblox 設定 レコードタイプセレクタ。

Ingress コントローラ

属性	タイプ	説明
pod_selector	オブジェクト	ポッドセレクタ
controller_config	オブジェクト	コントローラ設定

ポッドセレクタ

属性	タイプ	説明
名前空間	string	Ingress コントローラポッドが実行されている名前空間。

属性	タイプ	説明
ラベル	配列	Ingress コントローラポッドのラベルを指定する {“key”, “value”} ペアの配列。

コントローラ設定

属性	タイプ	説明
ingress_class	string	Ingress コントローラが満たす入力クラスの名前。
名前空間	string	名前空間は、Ingress コントローラが満たす名前空間の名前です。
http_ports	配列	HTTP ポートの配列。
https_ports	配列	HTTPS ポートの配列。

Infoblox 設定

enable_network_record	bool	デフォルト値は true です。false の場合、ネットワークタイプのレコードは無効になります。
enable_host_record	bool	デフォルト値は true です。false の場合、ホストタイプのレコードは無効になります。
enable_a_record	bool	デフォルト値は true です。false の場合、*A* タイプのレコードは無効になります。
enable_aaaa_record	bool	デフォルト値は true です。false の場合、*AAAA* タイプのレコードは無効になります。

** オーケストレータオブジェクトの読み取り専用ステータスフィールド **

属性	タイプ	説明
authentication_failure	bool	Secure Workload オーケストレータへの接続ステータス：*true* はオーケストレータへの接続が成功したことを示します。このフィールドが *false* の場合、*authentication_failure_error* フィールドには、接続失敗の理由を説明する詳細なエラーメッセージが表示されます。
authentication_failure_error	string	オーケストレータとの接続エラーやクレデンシャルエラーのデバッグに役立つ詳細なエラーメッセージ
scope_id	string	インベントリが公開および表示されるテナントルート範囲 ID

オーケストレーションの取得

このエンドポイントは、Secure Workload アプライアンスによって認識されているオーケストレータのリストを返します。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET/openapi/v1/orchestrator/{scope}
```

パラメータ：（なし）

指定されたルート範囲のオーケストレータ オブジェクトのリストを返します。*scope* はルート範囲 ID にする必要があります。

オーケストレーションの作成

このエンドポイントは、新しいオーケストレーションを作成するために使用されます。

```
POST/openapi/v1/orchestrator/{scope}
```

vCenter オーケストレータのサンプル Python コード

```
req_payload = {

"name": "VCenter Orchestrator"

"type": "vcenter",

"hosts_list": [ { "host_name": "8.8.8.8", "port_number": 443}],
```

```
"username":"admin",
"password":"admin"
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))
```

DNS オーケストレータのサンプル Python コード

```
req_payload = {
"name": "DNS Server"
"type": "dns",
"hosts_list": [ { "host_name": "8.8.8.8", "port_number": 53}],
"dns_zones": [ "lab.corp.com.", "dev.corp.com." ]
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))
```

Kubernetes オーケストレータのサンプル Python コード

```
req_payload = {
"name": "k8s"
"type": "kubernetes",
"hosts_list": [ { "host_name": "8.8.8.8", "port_number": 53}],
"certificate": "",
"key": "",
"ca_certificate": "",
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))
```

Ingress コントローラを使用した Kubernetes オーケストレータのサンプル Python コード

認証の詳細を作成するには、Kubernetes/OpenShift 外部オーケストレータに関する情報を参照してください。

```
req_payload = {
"name": "k8s"
"type": "kubernetes",
"hosts_list": [ { "host_name": "8.8.8.8", "port_number": 53}],
"certificate": "",
"key": "",
"ca_certificate": "",
"ingress_controllers": [
{
```

```
"pod_selector": {
"namespace": "ingress-nginx",
"labels": [{ "key": "app", "value": "nginx-ingress"}],
}
}
]
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))
```

複数の Ingress コントローラを使用した Kubernetes オーケストレータのサンプル Python コード

認証の詳細を作成するには、Kubernetes/OpenShift 外部オーケストレータに関する情報を参照してください。

```
req_payload = {
"name": "k8s"
"type": "kubernetes",
"hosts_list": [ { "host_name": "8.8.8.8", "port_number": 53}],
"certificate": "",
"key": "",
"ca_certificate": "",
"ingress_controllers": [
{
"pod_selector": {
"namespace": "ingress-nginx",
"labels": [{ "key": "app", "value": "nginx-ingress"}],
},
"controller_config": {
"ingress_class": "nginx-class",
}
},
{
"pod_selector": {
"namespace": "ingress-haproxy",
"labels": [{ "key": "app", "value": "haproxy-ingress"}],
},
"controller_config": {
```

```
"ingress_class": "haproxy-class",
"http_ports": [8080],
"https_ports": [8443],
"namespace": "haproxy-watching-namespace"
}
}
],
}
resp = restclient.post('/orchestrator/Default', json_body=json.dumps(req_payload))
** AWS および EKS タイプは、外部オーケストレータでサポートされなくなりました。これらのタイプは
コネクタに移植
されました。
```

特定のオーケストレーションを取得する

このエンドポイントは、オーケストレーションのインスタンスを返します。

`GET/openapi/v1/orchestrator/{scope}/{orchestrator_id}`

指定された ID に関連付けられているオーケストレーション オブジェクトを返します。

オーケストレーションの更新

このエンドポイントはオーケストレーションを更新します。

`PUT/openapi/v1/orchestrator/{scope}/{orchestrator_id}`

パラメータ：

POST パラメータと同じ

特定のオーケストレーションの削除

このエンドポイントは、指定されたオーケストレーションを削除します。

`DELETE /openapi/v1/orchestrator/{scope}/{orchestrator_id}`

オーケストレータのゴールデンルール

この一連の API を使用して、外部 Kubernetes オーケストレータのゴールデンルールを管理できます。ゴールデンルールは、許可リスト強制モードで Kubernetes コントロールプレーンの接続を確保するために必要です。API キーに関連付けられた `external_integration` 機能が必要です。

ゴールデンルールで現在サポートされているオーケストレータタイプは「kubernetes」のみです。Kubernetes 以外のオーケストレータに対するこのエンドポイントへの要求は失敗します。

オーケストレータのゴールデン ルール オブジェクト

オーケストレーション オブジェクトの属性については、以下で説明します。

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

オーケストレータのゴールデンルールを取得する

このエンドポイントは、関連付けられたゴールデンルールをオーケストレータに返します。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/orchestrator/{scope}/{id}/gr
```

パラメータ：（なし）

単一のゴールデン ルール オブジェクトを返します。

ゴールデンルールの作成/更新

このエンドポイントは、既存のオーケストレータのゴールデンルールを作成または更新するために使用されます。

```
POST /openapi/v1/orchestrator/{scope}/{id}/gr
```

パラメータ：

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

サンプル python コード

```
req_payload = {

"kubelet_port":10255,

"services": [
```

```
{
"説明": "kube-dns",
"addresses": [ "10.0.1.1:53/TCP", "10.0.1.1:53/UDP" ],
"consumed_by": [ "NODES", "PODS"],
}
]
}
resp = restclient.post('/orchestrator/{scope_id}/{orchestrator_id}/gr', json_
‹→body=json.dumps(req_payload))
```

FMC オーケストレータドメイン

この一連の API を使用して、外部 FMC オーケストレータのドメインを管理できます。特定の FMC ドメインでの適用を有効にするには、FMC ドメインが必要です。API キーに関連付けられた `external_integration` 機能が必要です。

現在サポートされている FMC ドメインのオーケストレータタイプは「fmc」のみです。non-fmc orches- trators でのこのエンドポイントへのリクエストは失敗します。

オーケストレータ FMC ドメインオブジェクト

オーケストレーション オブジェクトの属性については、以下で説明します。

属性	タイプ	説明
fmc_domains	配列	FMC ドメインオブジェクトの配列

FMC ドメインオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
name	string	FMC ドメインの名前
enforcement_enabled	boolean	このフラグはデフォルトで falseに設定されています。trueの場合、外部オーケストレータは、ワークスペースに対してポリシーの適用が実行されるときに、「name」に一致するドメインにポリシーを展開します。

URL 属性については、以下で説明します。

属性	タイプ	説明
scope	string	インベントリが公開および表示されるテナントルート範囲の名前または ID
orchestrator_id	string	FMC オーケストレータのオーケストレータ ID

FMC ドメインの取得

このエンドポイントは、FMC オーケストレータに関連付けられている FMC で構成されている fmc ドメインを返します。この API は、 `external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/orchestrator/{scope}/{orchestrator_id}/fmcdomains
```

パラメータ：（なし）

FMC ドメインオブジェクトの属性のリストを含む json オブジェクトを返します。

FMC 外部オーケストレータの FMC ドメイン設定の更新

このエンドポイントは、既存の FMC 外部オーケストレータの FMC ドメイン属性を更新します。

```
PUT /openapi/v1/orchestrator/{scope}/{orchestrator_id}/fmcdomains
```

パラメータ：

属性	タイプ	説明
fmc_domains	配列	FMC ドメインオブジェクトの配列

FMC ドメインオブジェクトの属性については、以下で説明します。

属性	タイプ	説明
name	string	FMC ドメインの名前

属性	タイプ	説明
enforcement_enabled	boolean	このフラグはデフォルトで falseに設定されています。trueの場合、外部オーケストレータは、ワークスペースに対してポリシーの適用が実行されるときに、「name」に一致するドメインにポリシーを展開します。

URL 属性については、以下で説明します。

属性	タイプ	説明
scope	string	インベントリが公開および表示されるテナントルート範囲の名前または ID
orchestrator_id	string	FMC オーケストレータのオーケストレータ ID

サンプル python コード

```
req_payload = {
"fmc_domains": [
{
"enforcement_enabled": False,
"name": "Global/Eng"
},
{
"enforcement_enabled": True,
"name": "Global/Prod"
}
]
}
resp = restclient.put('/orchestrator/{scope}/{orchestrator_id}/fmcdomains', json_
‹→body=json.dumps（req_payload））
```

RBAC（役割ベースのアクセス制御）に関する考慮事項

ルート範囲下でオーケストレータにアクセスするには、要求に使用されるAPIキーに要求の権限が必要です。すべてのオーケストレータ API 呼び出しは範囲が指定され、URL の一部として常にルート範囲 ID が必要です。オーケストレータは常にルート範囲レベルに存在し、サブ範囲の下に作成することはできません。特定のテナントルート範囲の下で作成されたオーケストレータ（およびこれらのオーケストレータによって学習されたインベントリ）は、他のテナントからは見えません。

複数のルートドメイン（vrfs）が構成されている可能性がある F5 ロードバランサの場合、F5 ルート ドメイン フィルタリング ロジックは、すべてのパーティションにわたって F5 上のすべてのエンティティをスキャンしますが、 F5 オーケストレータの *route_domain* フィールドで指定されたルートドメインとして評価されないエンティティ （サービス、snat プール、プール、およびバックエンド）は破棄します。

高可用性とフェールオーバーに関する考慮事項

hosts_listパラメータを使用すると、オーケストレータに複数のサーバーアドレスを設定できます。複数のサーバーアドレスの場合の Secure Workload サーバー選択ロジックは、オーケストレータタイプごとに異なります。

vCenter、Kubernetes、DNS、F5、NetScaler、Infoblox の場合、選択は最初の正常なエンドポイントに基づいて行われます。接続は永続的ではないため（Kubernetesを除く）、Secure Connector Orchestrator Managerは、ポーリング周期ごとにホストをスキャンし、hosts_listで検出された最初の正常なエンドポイントをポーリングします。Kubernetesの場合、永続的なイベントチャネルが維持され、接続が失敗すると、次の正常なエンドポイントを使用して、すべてのホストのスキャンと後続の完全なポーリングが実行されます。

Kubernetes RBAC リソースに関する考慮事項

Kubernetes クライアントは、次のリソースの GET/LIST/WATCH を試みます。

指定するKubernetes認証の資格情報には、次のリソースに対する最小限の権限セットが必要です。

リソース	**Verbs**
daemonsets	[get list watch]
deployments	[get list watch]
endpoints	[get list watch]
名前空間	[get list watch]

リソース	Verbs
ノード	[get list watch]
ポッド	[get list watch]
replicasets	[get list watch]
replicationcontrollers	[get list watch]
サービス	[get list watch]
statefulsets	[get list watch]
daemonsets.apps	[get list watch]
deployments.apps	[get list watch]
endpoints.apps	[get list watch]
namespaces.apps	[get list watch]
nodes.apps	[get list watch]
pods.apps	[get list watch]
replicasets.apps	[get list watch]
replicationcontrollers.apps	[get list watch]
services.apps	[get list watch]
statefulsets.apps	[get list watch]
daemonsets.extensions	[get list watch]
deployments.extensions	[get list watch]
endpoints.extensions	[get list watch]
namespaces.extensions	[get list watch]
nodes.extensions	[get list watch]
pods.extensions	[get list watch]
replicasets.extensions	[get list watch]
replicationcontrollers.extensions	[get list watch]
services.extensions	[get list watch]
statefulsets.extensions	[get list watch]

サイト情報

このAPIを使用して、クラスタの状態、クラスタの種類、外部IP、電子メールなどのクラスタ情報を取得できます。

（注） この API は、サイト管理者ユーザーのみが使用できます。

サイト情報の取得

このエンドポイントは、クラスのタサイト情報を持つ JSON オブジェクトを返します。

```
GET /openapi/v1/site_infos
```

パラメータ：（なし）

応答オブジェクト：クラスタのサイト情報を持つ JSON オブジェクト

サンプル python コード

```
resp = restclient.get('/site_infos')
```

サンプル応答

```
{
"cluster_state": "Enabled till 2020-12-31 23:59:59 UTC",
"cluster_uuid": "00000000-0000-0000-0000-000000000000",
"site_bosun_email": "customer-support@company.com",
"site_cluster_type": "physical",
"site_external_ips": [
"1.1.1.1",
"1.1.1.2",
...
"1.1.1.7"
],
"site_name": "cluster_name",
"site_sensor_vip_ip": "2.1.1.1",
"site_ui_admin_email": "site-admin@company.com",
(
"site_ui_fqdn": "cluster.company.com",
"site_ui_primary_customer_support_email": "customer-support@company.com"
```

```
}
```

クラスタの正常性

この API を使用して、Cisco Secure Workload 内のすべての物理サーバーのステータスを取得できます。

（注） この API は、サイト管理者ユーザーのみが使用できます。

クラスタ正常性の取得

このエンドポイントは、クラスタ正常性情報を含む JSON オブジェクトを返します。

```
GET /openapi/v1/cluster_nodes
```

パラメータ：（なし）

応答オブジェクト： クラスタ正常性情報を含む JSON オブジェクト

サンプル python コード

```
resp = restclient.get('/cluster_nodes')
```

Service Health

この API を使用して、Secure Workload クラスタで使用される全サービスの正常性をサービスの依存関係とともに取得できます。

（注） この API は、サイト管理者ユーザーのみが使用できます。

サービス正常性の取得

このエンドポイントは、サービス正常性情報を含む JSON オブジェクトを返します。

```
GET /openapi/v1/service_status
```

パラメータ：（なし）

応答オブジェクト： サービス正常性情報を含む JSON オブジェクト

サンプル python コード

```
resp = restclient.get('/service_status')
```

Secure Connector

OpenAPI は、Secure Connector の機能を管理するエンドポイントを公開します。これらのエンドポイントでは、`external_integration` 機能が API キーに関連付けられている必要があります。

（注） Secure Connector API は、サイトレベルでは使用できません。これらはルート範囲レベルでのみ使用できます。

ステータスの取得

このエンドポイントは、指定されたルート範囲の Secure Connector トンネルの現在のステータスを返します。

```
GET /openapi/v1/secureconnector/name/{ROOT_SCOPE_NAME}/status
```

```
GET /openapi/v1/secureconnector/{ROOT_SCOPE_ID}/status
```

指定されたルート範囲に対する読み取りアクセス許可が必要です。

返されるステータスは、次のスキーマを持つ JSON オブジェクトです。

キー	タイプ	値
active	boolean	Secure Connector トンネルは現在アクティブです。
peer	string	トンネルの Secure Connector クライアント側の <ip>:<port>
start_time	int	トンネルが開始された時のタイムスタンプ（秒単位のエポック時間）
last_heartbeat	int	クライアントから最後に受信したハートビートのタイムスタンプ（秒単位のエポック時間）

トークンを取得

このエンドポイントは、指定されたルート範囲の Secure Connector クライアントをブートストラップするために使用される新しい 1 回限りの期間限定トークンを返します。

```
GET /openapi/v1/secureconnector/name/{ROOT_SCOPE_NAME}/token
```

```
GET /openapi/v1/secureconnector/{ROOT_SCOPE_ID}/token
```

指定されたルート範囲に対する OWNER 権限が必要です。

返されるトークンは、暗号で署名されたトークンを含む文字列で、1時間有効です。有効なトークンは、Secure Connector クライアントをブートストラップするために1回だけ使用できます。

証明書のローテーション

このエンドポイントは、指定されたルート範囲の新しい証明書の作成を強制的に実行します。新しい証明書は Secure Connector サーバーによって使用され、このルート範囲のクライアントからの証明書署名要求に署名するために使用されます。

```
POST /openapi/v1/secureconnector/name/{ROOT_SCOPE_NAME}/rotate_certs?invalidate_old=
‹→{true|false}

POST /openapi/v1/secureconnector/{ROOT_SCOPE_ID}/rotate_certs?invalidate_old=
‹→{true|false}
```

指定されたルート範囲に対する OWNER 権限が必要です。

このエンドポイントが呼び出されると、このルート範囲のクライアントとサーバー間の通信は、すぐに新しい証明書を使用するように移行します。

invalidate_old が false に設定されている場合、既存のクライアントは新しい公開キー/秘密キーのペアを自動的に作成し、既存の証明書を使用して、新しい公開キーの新しい証明書に署名します。

invalidate_old が true に設定されている場合、既存の証明書はすぐに無効になります。既存のクライアントはサーバーに接続できなくなり、新しいトークンを使用して再度ブートストラップする必要があります。詳細については、「Secure Connector の展開」を参照してください。

外部オーケストレータのポリシー適用ステータス

この一連の API は、*F5 BIG-IP* や *Citrix Netscaler* などのロードバランサ外部オーケストレータにポリシー適用ステータスを提供するために使用されます。

（注） これらの API を使用するには、VRF にアタッチされた範囲にユーザーがアクセスできる必要があります。

すべての外部オーケストレータのポリシー適用ステータスを取得する

このエンドポイントは、指定された VRF に属するすべての外部オーケストレータのポリシー適用ステータスを返します。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/tnp_policy_status/{vrfID}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
vrfID	integer	ルート範囲の VRF ID。

応答オブジェクト：ステータスが ENFORCED または FAILED または IGNORED のネットワークポリシーのリストを返します。

サンプル python コード

```
vrf_id = 676767

restclient.get('/tnp_policy_status/%d' % vrf_id)
```

外部オーケストレータのポリシー適用ステータスの取得

このエンドポイントは、指定されたVRFに属する外部オーケストレータのポリシー適用ステータスを返します。この API は、external_integration 機能を備えた API キーで使用できます。

```
GET /openapi/v1/tnp_policy_status/{vrfID}/{orchestratorID}
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
vrfID	integer	ルート範囲の VRF ID。
orchestratorID	string	外部オーケストレータの ID。

応答オブジェクト：ステータスが ENFORCED または FAILED または IGNORED のネットワークポリシーのリストを返します。

サンプル python コード

```
vrf_id = 676767

orchestrator_id = '5ee3c991497d4f3b00f1ee07'

restclient.get('/tnp_policy_status/%d/%s' % (vrf_id, orchestrator_id))
```

管理対象データタップとデータシンクの証明書のダウンロード

この一連の API は、管理対象データタップとデータシンクの証明書をダウンロードするために使用されます。

（注） これらの API を使用するには、VRF にアタッチされた範囲にユーザーがアクセスできる必要があります。

特定の VRF ID に対するマネージド DataTap のリストの取得

このエンドポイントは、指定された VRF のマネージドデータタップのリストを返します。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/mdt/{vrfID}
```

パラメータ：（なし）

マネージドデータタップ ID などの属性を持つマネージドデータタップのリストを返します。

指定された MDT ID に対するマネージド DataTap 証明書のダウンロード

このエンドポイントは、指定されたマネージドデータタップ ID の証明書をダウンロードするために使用されます。MDT ID は、上記のドキュメントで説明されているように、**/openapi/v1/mdt/{vrfID}** エンドポイントを使用して取得できます。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/mdt/{vrfID}/{mdtID}/certs
```

パラメータ：

名前	タイプ	説明
format	string	キーストアと信頼ストアのフォーマット。値：`jks`（デフォルト値）または `cert`

次のファイルを含む tar.gz ファイルを返します。

`jks` 形式の場合：**truststore.jks**、**topic.txt**、**passphrase.txt**、**keystone.jks**、**kafkaBrokerIps.txt**、**consumer_name.txt**、**consumer_group_id.txt** .

`jks` 形式の場合：**KafkaConsumerCA.cert**、**KafkaConsumerPrivateKey.key**、**kafkaCA.cert**、**kafkaBrokerIps.txt**、**topic.txt**

KafkaConsumerCA.cert は公開証明書ファイルであり、**KafkaConsumerPrivateKey.key** ファイルには秘密キーがあります。**kafkaCA.cert** には CA 証明書があり、**kafkaBrokerIps.txt** には Kafka ブローカーの IP アドレスとポートのリストがあります。topic.txt ファイルには、MDT からデータを取得するために使用する必要があるトピックの名前が含まれています。**truststore.jks** および **keystone.jks** は、Java キーストアファイルです。

特定の VRF ID に対する DataSink のリストの取得

このエンドポイントは、指定された VRF の DataSink のリストを返します。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/datasinks/{vrfID}
```

パラメータ：（なし）

DataSink ID などの属性を持つ DataSink のリストを返します。

指定された DataSink ID の DataSink 証明書をダウンロードします。

このエンドポイントは、指定された DataSink ID の証明書をダウンロードするために使用されます。DataSink ID は、上記のドキュメントで説明されているように、**/openapi/v1/datasinks/{vrfID}** エンドポイントを使用して取得できます。この API は、`external_integration` 機能を備えた API キーで使用できます。

```
GET /openapi/v1/datasinks/{vrfID}/{dsID}/certs
```

パラメータ：（なし）

userCA.cert、**userPrivateKey.key**、**intermediateCA.cert**、**kafkaCA.cert**、**kafkaBrokerIps.txt**、**topic.txt** ファイルを含む tar.gz ファイルを返します。

userCA.cert は公開証明書ファイルであり、**KafkaConsumerPrivateKey.key** ファイルには秘密キーがあります。**intermediateCA. cert** および **kafkaCA.cert** にはそれぞれ中間 CA 証明書とルート CA 証明書があります。**kafkaBrokerIps.txt** には Kafka ブローカーの IP アドレスとポートのリストがあります。**topic.txt** ファイルには、DataSink からデータを取得するために必要なトピックの名前があります。

変更ログ

この API は、ログ項目を変更するための読み取りアクセスを提供します。この API には、API キーに関連付けられた `user_role_scope_management` 機能が必要です。

（注） この API は、サイト管理者とルート範囲の所有者のみが使用できます。

変更ログオブジェクト

変更ログオブジェクトの属性の説明は以下のとおりです。

属性	タイプ	説明
id	string	変更ログ項目の固有識別子。
association_chain	オブジェクトの配列	この変更に関連する名前と ID のリスト。
scope	string	変更の範囲（Secure Workload の範囲とは異なります）。
action	string	アクションを変更します。
details	string	利用可能な場合、追加アクションの詳細。

属性	タイプ	説明
created_At	整数	変更ログ項目が作成されたときの UNIX タイムスタンプ。
modifier	オブジェクト	変更を行うユーザー。
modified	オブジェクト	変更されたフィールドと値。
original	オブジェクト	変更前のフィールドと値。
version	整数	バージョン ID。

検索

このエンドポイントは、指定された条件に一致する変更ログのリストを返します。

```
GET /openapi/v1/change_logs
```

パラメータ：要求 URL には、次のパラメータが含まれています。

名前	タイプ	説明
root_app_scope_id	string	（オプション）ルート範囲の所有者に必要です。ルート範囲で結果をフィルタリングします。
association_name	string	（オプション）ルート範囲の所有者に必要です。指定した項目タイプを返します。例：「H4Users」
history_action	string	（オプション）アクションを変更します。例：「update」
details	string	（オプション）アクションの詳細。例：「soft-delete」
before_epoch	integer	（オプション）この UNIX タイムスタンプより前に作成された結果を含めます。
after_epoch	integer	（オプション）この UNIX タイムスタンプの後に作成された結果を含めます。
offset	integer	（オプション）スキップする結果の数。

名前	タイプ	説明
limit	integer	（オプション）結果の制限数。

応答オブジェクト：変更ログオブジェクトのリストを返します。

応答

この応答は、本文の JSON オブジェクトで、次のプロパティがあります。

名前	タイプ	説明
total_count	integer	オフセットまたは制限を適用する前に一致した項目の総数。
Items	オブジェクトの配列	結果のリスト。

サンプル python コード

過去 1 日以内の指定されたルート範囲内の最後の 100 件の範囲オブジェクトの変更を取得します。

```
root_app_scope_id = '5ce480db497d4f1ca1fc2b2b'

one_day_ago = int(time.time() - 24 * 60 * 60)

resp = restclient.get('/change_logs', params={'root_app_scope_id': root_app_scope_id,

'association_name': 'AppScope',

'after_epoch': one_day_ago,

'limit': 100})
```

これらの結果をさらに絞り込み、新しい範囲の作成のみを表示します。

```
root_app_scope_id = '5ce480db497d4f1ca1fc2b2b'

one_day_ago = int(time.time() - 24 * 60 * 60)

resp = restclient.get('/change_logs', params={'root_app_scope_id': root_app_scope_id,

'association_name': 'AppScope',

'history_action': 'create',

'after_epoch': one_day_ago,

'limit': 100})
```

サイト管理者は、制限とオフセットを使用して、すべての範囲にわたるすべての変更を繰り返し取得できます。

```
resp = restclient.get('/change_logs', params={'offset': 100, 'limit': 100})
```

ルーティング不可能なエンドポイント

この一連のAPIは、ルーティング不可能なエンドポイントを管理したり、IP/サブネットがルーティング不可能であることを示したり、ユーザーによってルーティング不可能とマークされたエンドポイントのリストを取得したり、IP/サブネットに付いたルーティング不可能なエンドポイントのマークを解除したりするために使用されます。API キーに関連付けられた `user_data_upload` 機能が必要です。

ルーティング不可能なエンドポイントオブジェクト

ルーティング不可能なエンドポイントオブジェクトの属性について、以下で説明します。

属性	タイプ	説明
id	string	ルーティング不可能なエンドポイントの一意の識別子。
name	string	ルーティング不可能なエンドポイントのユーザー指定の名前。
subnet	string	IPv4/IPv6 サブネット。
vrf-id	long	ルーティング不可能なエンドポイントが属する VRF の ID。
address_type	string	サブネットアドレスタイプに基づく IPv4/IPv6
host_uuid	string	エージェントの一意の ID
description	string	ルーティング不可能なエンドポイントのユーザー指定の説明。

ルーティング不可能なエンドポイントの取得

このエンドポイントは、指定されたテナント内のルーティング不可能なエンドポイントのリストを返します。

```
GET /openapi/v1/non_routable_endpoints/{rootScopeName}
```

パラメータ：（なし）

ルーティング不可能なエンドポイントの作成

このエンドポイントは、ルーティング不可能なエンドポイントを作成するために使用されます。

```
POST /openapi/v1/non_routable_endpoints/{rootScopeName}
```

パラメータ：

属性	タイプ	説明
name	string	ルーティング不可能なエンドポイントのユーザー指定の名前。
subnet	string	IPv4/IPv6 サブネット。
address_type（オプション）	string	サブネットアドレスタイプに基づく IPv4/IPv6
host_uuid（オプション）	string	エージェントの一意の ID
description（オプション）	string	ルーティング不可能なエンドポイントのユーザー指定の説明。

*オプションのフィールドが指定されていない場合、null 値が入力されます。

サンプル python コード

```
req_payload = {
"name": "nre-1",
"subnet": "1.1.1.1/30",
"address_type": IPV4,
"description": "sample parameters test"
}
resp = restclient.post('/openapi/v1/non_routable_endpoints/Default', json_body=json.
→dumps(req_payload))
```

名前を使用したルーティング不可能なエンドポイントの取得

このエンドポイントは、指定された名前のルーティング不可能なエンドポイントを返します。

```
GET /openapi/v1/non_routable_endpoints/{rootScopeName}/name/{name}
```

パラメータ：（なし）

ID を使用したルーティング不可能な特定のエンドポイントの取得

このエンドポイントは、指定された ID のルーティング不可能なエンドポイントを返します。

```
GET /openapi/v1/non_routable_endpoints/{rootScopeName}/id/{id}
```

パラメータ：（なし）

ルーティング不可能な特定のエンドポイント名の更新

このエンドポイントは、ルーティング不可能なエンドポイントを更新するために使用されます。既存のルーティング不可能なエンドポイントの ID または名前を使用して、エンドポイント名前を更新します。

```
PUT /openapi/v1/non_routable_endpoints/{rootScopeName}
```

パラメータ：

属性	タイプ	説明
id	string	ルーティング不可能なエンドポイントの一意の識別子。
name	string	ルーティング不可能なエンドポイントのユーザー指定の名前。
new_name	string	更新する新しい名前

サンプル python コード

```
req_payload = {

"name": "nre-1",

"new_name": "nre-updated",

}

resp = restclient.put('/openapi/v1/non_routable_endpoints/Default', json_body=json.

‹→dumps(req_payload))

req_payload = {

"id": "5f706964a5b5f16ed4b0aacb",

"new_name": "nre-updated",

}

resp = restclient.put('/openapi/v1/non_routable_endpoints/Default', json_body=json.

‹→dumps(req_payload))
```

名前を使用したルーティング不可能なエンドポイントの削除

このエンドポイントは、特定のルーティング不可能なエンドポイントを削除します。

```
DELETE /openapi/v1/non_routable_endpoints/{rootScopeName}/name/{name}
```

名前を使用したルーティング不可能なエンドポイントの削除

このエンドポイントは、特定のルーティング不可能なエンドポイントを削除します。

```
DELETE /openapi/v1/non_routable_endpoints/{rootScopeName}/id/{id}
```

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。