Ajoutez les propriétés de l'instance cloud de l'équipement via l'API REST

Publié: 2023-10-24

Les propriétés du cloud de l'appareil vous permettent de consulter les informations relatives à votre environnement cloud dans le système ExtraHop. Vous pouvez identifier le nom, le type et l'ID de l'instance cloud d'un équipement, ainsi que le compte cloud propriétaire de l'équipement et l'identifiant du cloud privé virtuel dans lequel se trouve l'équipement.

Ce guide fournit des instructions pour ajouter une observation via l'explorateur d'API ExtraHop et des exemples de scripts Python pour Amazon AWS et Microsoft Azure. Si vous mettez à jour les propriétés du cloud à l'aide d'un script d'API REST, vous pouvez récupérer en permanence des informations auprès de votre fournisseur de cloud pour vous assurer que les informations relatives à vos propriétés cloud sont toujours à jour.

Avant de commencer

- Vous devez vous connecter au sonde ou console avec un compte disposant de tous les privilèges d'écriture nécessaires pour générer une clé d'API.
- Vous devez disposer d'une clé d'API valide pour apporter des modifications via l'API REST et suivre les procédures ci-dessous. (Voir Génération d'une clé d'API.)
- Familiarisez-vous avec Guide de l'API REST ExtraHop pour apprendre à naviguer dans l' explorateur d'API ExtraHop.

Ajoutez des propriétés d'instance cloud via l'explorateur d'API ExtraHop

Important: L'explorateur d'API REST n'est pas disponible sur Reveal (x) 360.

1. Dans un navigateur, accédez à l'explorateur d'API ExtraHop.

   L'URL est le nom d'hôte ou l'adresse IP de votre sonde ou console, suivi de `/api/v1/explore/`. Par exemple, si votre nom d'hôte est seattle-eda, l'URL est `https://seattle-eda/api/v1/explore/`.
2. Cliquez **Entrez la clé d'API** puis collez ou saisissez votre clé d'API dans le **Clé d'API** champ.
3. Cliquez **Autoriser** puis cliquez sur **Fermer**.
4. Trouvez l'ID de l'équipement en recherchant son adresse MAC.
   a) Cliquez **Appareil** puis cliquez sur **POST /appareils/recherche**.
   b) Cliquez **Essayez-le**.
   c) Dans le champ body, spécifiez le code JSON suivant, en remplaçant MACADDRESS par l'adresse MAC de votre équipement cloud :

   ```
   {
     "filter": {
       "field": "macaddr",
       "operand": "MACADDRESS",
       "operator": "="
     }
   }
   ```

   d) Cliquez **Envoyer une demande**.
   e) Dans la section Corps de la réponse, visualisez et enregistrez la valeur du `id` champ pour chaque équipement renvoyé.
5. Ajoutez les métadonnées de l'équipement cloud.
   a) Cliquez **PATCH /appareils/ {id}**.
   b) Cliquez **Essayez-le**.

c) Dans le `id` champ, spécifiez un ID.

d) Dans le champ body, spécifiez le code JSON suivant, en remplaçant le `string` valeurs associées aux propriétés de votre environnement cloud :

```
{
  "cloud_account": "string",
  "cloud_instance_id": "string",
  "cloud_instance_name": "string",
  "cloud_instance_type": "string",
  "vpc_id": "string"
}
```

e) Cliquez **Envoyer une demande**.

Récupérez et installez l'exemple de script Python Lambda pour AWS

Le référentiel GitHub d'ExtraHop contient un exemple de script Python qui importe les propriétés d'une instance AWS EC2 dans le système ExtraHop. Le script mappe les interfaces réseau des instances EC2 aux appareils découverts sur le système ExtraHop par adresse MAC.

Le script est conçu pour s'exécuter en tant que fonction Lambda dans AWS. Voici quelques points importants à prendre en compte pour exécuter le script dans AWS :

- Le script est conçu pour s'exécuter sur un intervalle de temps défini. Chaque fois que le script est exécuté, il analyse chaque instance du VPC et met à jour les appareils correspondants dans le système ExtraHop. Pour plus d'informations sur la configuration d'une fonction Lambda pour une exécution périodique, consultez le didacticiel AWS ici.
- La fonction Lambda doit pouvoir accéder aux ressources de votre VPC. Pour plus d' informations, consultez le didacticiel AWS ici.
- La fonction Lambda doit disposer d'un accès en liste et en lecture à l' action DescribeInstances pour le service EC2. Pour plus d'informations, consultez le didacticiel AWS ici.

Important: L'exemple de script python s'authentifie auprès de la sonde ou de la console via une clé API, qui n'est pas compatible avec l' API REST Reveal (x) 360. Pour exécuter ce script avec Reveal (x) 360, vous devez modifier le script pour vous authentifier à l'aide de jetons d'API. Consultez le py_rx360_auth.py script dans le référentiel GitHub ExtraHop pour un exemple de procédure d'authentification à l'aide de jetons API.

Note: Si le script renvoie un message d'erreur indiquant que la vérification du certificat SSL a échoué, assurez-vous que un certificat de confiance a été ajouté à votre sonde ou à votre console. Vous pouvez également ajouter le `verify=False` possibilité de contourner la vérification des certificats. Cependant, cette méthode n'est pas sûre et n'est pas recommandée. Le code suivant envoie une requête HTTP GET sans vérification de certificat :

```
requests.get(url, headers=headers, verify=False)
```

1. Accédez à l'ExtraHop référentiel GitHub d'exemples de code et téléchargez le `add_cloud_props_lambda/add_cloud_props_lambda.py` fichier sur votre machine locale.
2. Dans un éditeur de texte, ouvrez le `add_cloud_props_lambda.py` archivez et remplacez les variables de configuration suivantes par des informations provenant de votre environnement :
   - **NOM D'HÔTE:** L'adresse IP privée ou le nom d'hôte de l'instance EC2 de la sonde ou de la console.
   - **APIKEY:** La clé API ExtraHop.
3. Ajoutez le `add_cloud_props_lambda.py` fichier dans un fichier zip avec le `requests` module Python.

Le script importe le `requests` Module Python, qui n'est pas disponible pour les fonctions Lambda par défaut. Pour plus d'informations sur la création d'un fichier zip pour importer des bibliothèques tierces dans Lambda, consultez le Documentation AWS.

4. Dans AWS, créez une fonction Lambda.

   Pour plus d'informations sur la création de fonctions Lambda, consultez le Documentation AWS.
5. Sur la page de la fonction Lambda, cliquez sur **Actions** et sélectionnez **Téléchargez un fichier .zip** fichier.
6. Sélectionnez le fichier zip que vous avez créé.

Récupérez et installez l'exemple de script Python pour Azure

Le référentiel GitHub d'ExtraHop contient un exemple de script Python qui importe les propriétés de l'équipement Azure dans le système ExtraHop. Le script attribue des propriétés d'équipement cloud à chaque équipement découvert par le système ExtraHop avec une adresse MAC appartenant à une interface réseau Azure VM. Le script est conçu pour être exécuté à un intervalle de temps défini. Chaque fois que le script est exécuté, il analyse chaque machine virtuelle et met à jour les appareils correspondants dans ExtraHop.

Le script nécessite les modules suivants du SDK Azure Python :

- azure.mgmt.compute
- azure.mgmt.network
- azure.common.credentials

Le script nécessite également que vous ayez configuré les identifiants d'authentification Azure dans les variables d'environnement suivantes sur la machine qui exécute le script :

- AZURE_SUBSCRIPTION_ID
- AZURE_CLIENT_ID
- AZURE_CLIENT_SECRET
- AZURE_TENANT_ID

Pour plus d'informations sur la génération de ces informations d'identification, consultez le Documentation Azure.

Important: L'exemple de script python s'authentifie auprès de la sonde ou de la console via une clé API, qui n'est pas compatible avec l' API REST Reveal (x) 360. Pour exécuter ce script avec Reveal (x) 360, vous devez modifier le script pour vous authentifier à l'aide de jetons d'API. Consultez le py_rx360_auth.py script dans le référentiel GitHub ExtraHop pour un exemple de procédure d'authentification à l'aide de jetons API.

1. Accédez au Référentiel GitHub d'exemples de code ExtraHop et téléchargez le `add_cloud_props_azure/add_cloud_props_azure.py` fichier sur votre machine locale.
2. Dans un éditeur de texte, ouvrez `add_cloud_props_azure.py` archivez et remplacez les variables de configuration suivantes par des informations provenant de votre environnement :

   • **NOM D'HÔTE:** L'adresse IP ou le nom d'hôte de la sonde ou de la console.

   • **APIKEY:** La clé API ExtraHop.
3. Exécutez la commande suivante :

```
python3 add_cloud_props_azure.py
```

Note: Si le script renvoie un message d'erreur indiquant que la vérification du certificat SSL a échoué, assurez-vous que un certificat de confiance a été ajouté à votre sonde ou à votre console. Vous pouvez également ajouter le `verify=False` possibilité de contourner la vérification des certificats. Cependant, cette méthode n'est pas sûre et

n'est pas recommandée. Le code suivant envoie une requête HTTP GET sans vérification de certificat :

```
requests.get(url, headers=headers, verify=False)
```