Guide du développeur

# AWS Panorama

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# AWS Panorama: Guide du développeur

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d’Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n’est pas d’Amazon, d’une manière susceptible de créer une confusion parmi les clients, ou d’une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d’Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

# Table of Contents

# Qu'est-ce que AWS Panorama ?

AWS Panoramaest un service qui intègre la vision par ordinateur à votre réseau de caméras sur site. Vous installez le pluginAWS PanoramaAppliance ou autre appareil compatible dans votre centre de données, enregistrez-le auprès deAWS Panorama, et déployez des applications de vision par ordinateur à partir du cloud.AWS Panoramafonctionne avec vos caméras réseau RTSP (Real Time Streaming Protocol) existantes. L'appliance exécute des applications de vision par ordinateur sécurisées à partir deAWSPartenaires, ou des applications que vous créez vous-même à l'aide deAWS PanoramaSDK d'application.

LeAWS PanoramaAppliance est un appareil périphérique compact qui utilise un puissant system-on-module (SOM) qui est optimisé pour les charges de travail d'apprentissage automatique. L'appareil peut exécuter plusieurs modèles de vision artificielle sur plusieurs flux vidéo en parallel et afficher les résultats en temps réel. Il est conçu pour une utilisation dans des environnements commerciaux et industriels et est classé pour la protection contre la poussière et les liquides (IP-62).

LeAWS PanoramaAppliance vous permet d'exécuter des applications de vision par ordinateur autonomes à la périphérie, sans envoyer d'images vers le cloud AWS. En utilisant le kit SDK AWS, vous pouvez intégrer d'autres services AWS et les utiliser pour suivre les données de l'application au fil du temps. En intégrant d'autres services AWS, vous pouvez utiliserAWS Panoramapour effectuer les opérations suivantes :

- Analyse des modèles de trafic— Utilisez le kit SDK AWS pour enregistrer des données à des fins d'analyse de vente au détail dans Amazon DynamoDB. Utilisez une application sans serveur pour analyser les données collectées au fil du temps, détecter les anomalies dans les données et prévoir le comportement future.
- Recevez des alertes de sécurité sur site— Surveiller les zones interdites sur un site industriel. Lorsque votre application détecte une situation potentiellement dangereuse, chargez une image sur Amazon Simple Storage Service (Amazon S3) et envoyez une notification à une rubrique Amazon Simple Notification Service (Amazon SNS) afin que les destinataires puissent prendre des mesures.
- Améliorer le contrôle de qualitésurveillez la sortie d'une chaîne de montage pour identifier les pièces qui ne sont pas conformes aux exigences. Mettez en surbrillance les images de pièces non conformes avec du texte et un cadre de sélection et affichez-les sur un écran pour examen par votre équipe de contrôle qualité.

- Collectez des données d'entraînement et— Téléchargez des images d'objets que votre modèle de vision par ordinateur n'a pas pu identifier, ou pour lesquels la confiance du modèle dans sa supposition était limite. Utilisez une application sans serveur pour créer une file d'attente d'images qui doivent être balisées. Étiquetez les images et utilisez-les pour réentraîner le modèle sur Amazon SageMaker.

AWS Panoramautilise d'autres services AWS pour gérerAWS PanoramaAppliance, accès aux modèles et au code, et déploiement d'applications.AWS Panoramafait autant que possible sans que vous ayez à interagir avec d'autres services, mais la connaissance des services suivants peut vous aider à comprendre commentAWS Panoramaest.

- SageMaker— Vous pouvez utiliser SageMaker pour collecter des données d'entraînement à partir de caméras ou de capteurs, créer un modèle d'apprentissage automatique et le former à la vision artificielle.AWS Panoramales usages SageMaker Neo pour optimiser les modèles afin qu'ils s'exécutent surAWS PanoramaAppliance.
- Amazon S3— Vous utilisez les points d'accès Amazon S3 pour préparer le code d'application, les modèles et les fichiers de configuration en vue d'un déploiement sur unAWS PanoramaAppliance.
- AWS IoT–AWS Panoramales usagesAWS IoTservices pour surveiller l'état deAWS PanoramaAppliance, gestion des mises à jour logicielles et déploiement d'applications. Vous n'avez pas besoin d'utiliserAWS IoTdirectement.

Pour commencer à utiliser le pluginAWS Panoramaet en savoir plus sur le service, passez àDémarrer avec AWS Panorama.

# Démarrer avec AWS Panorama

Pour commencerAWS Panorama, découvrez d'abord les concepts du service et la terminologie utilisée dans ce guide. Vous pouvez ensuite utiliser laAWS Panorama console pour enregistrer votreAWS Panorama appareil et créer une application. En une heure environ, vous pouvez configurer l'appareil, mettre à jour son logiciel et déployer un exemple d'application. Pour suivre les didacticiels de cette section, vous devez utiliser l'AWS Panoramaappliance et une caméra qui diffuse des vidéos sur un réseau local.

> **Note**
>
> Pour acheter unAWS Panorama appareil, rendez-vous sur laAWS Panorama console.

L'AWS Panoramaexemple d'application montre l'utilisation desAWS Panorama fonctionnalités. Il inclut un modèle qui a été entraîné SageMaker et un exemple de code qui utilise le SDK de l'AWS Panoramaapplication pour exécuter des inférences et générer des vidéos. L'exemple d'application inclut unAWS CloudFormation modèle et des scripts qui montrent comment automatiser les flux de travail de développement et de déploiement à partir de la ligne de commande.

Les deux dernières rubriques de ce chapitre détaillent les exigences relatives aux modèles et aux caméras, ainsi que les spécifications matérielles de l'AWS Panoramaappliance. Si vous n'avez pas encore acheté d'appareil et de caméras, ou si vous envisagez de développer vos propres modèles de vision par ordinateur, consultez d'abord ces rubriques pour plus d'informations.

Rubriques

# Concepts d'AWS Panorama

Dans AWS Panorama, vous créez des applications de vision par ordinateur et vous les déployez sur l'appliance AWS Panorama ou sur un appareil compatible pour analyser les flux vidéo provenant de caméras réseau. Vous écrivez du code d'application en Python et vous créez des conteneurs d'applications avec Docker. Vous utilisez l'interface de ligne de commande de l'application AWS Panorama pour importer des modèles d'apprentissage automatique localement ou depuis Amazon Simple Storage Service (Amazon S3). Les applications utilisent le kit SDK d'applications AWS Panorama pour recevoir l'entrée vidéo d'une caméra et interagir avec un modèle.

Concepts

- L'appliance AWS Panorama
- Appareils compatibles
- Applications
- Nœuds
- Modèles

## L'appliance AWS Panorama

L'appliance AWS Panorama est le matériel qui exécute vos applications. Vous utilisez la console AWS Panorama pour enregistrer une appliance, mettre à jour son logiciel et y déployer des applications. Le logiciel de l'appliance AWS Panorama se connecte aux flux de caméras, envoie des images vidéo à votre application et affiche la sortie vidéo sur un écran connecté.

L'appliance AWS Panorama est uneappareil Edge propulsé par Nvidia Jetson AGX Xavier. Au lieu d'envoyer des images auAWSCloud pour le traitement, il exécute les applications localement sur du matériel optimisé. Cela vous permet d'analyser la vidéo en temps réel et de traiter les résultats localement. L'appliance a besoin d'une connexion Internet pour signaler son état, télécharger des journaux et effectuer des mises à jour logicielles et des déploiements.

Pour plus d'informations, veuillez consulter Gestion duAWS PanoramaAppliance.

## Appareils compatibles

Outre l'appliance AWS Panorama, AWS Panorama prend en charge les appareils compatibles deAWSLes partenaires. Les appareils compatibles prennent en charge les mêmes fonctionnalités

que l'appliance AWS Panorama. Vous enregistrez et gérez les appareils compatibles avec la console et l'API AWS Panorama, et vous créez et déployez des applications de la même manière.

- Lenovo ThinkEdge® SE70— Propulsé par Nvidia Jetson Xavier NX

Le contenu et les exemples d'applications de ce guide sont développés avec l'appliance AWS Panorama. Pour plus d'informations sur les fonctionnalités matérielles et logicielles spécifiques de votre appareil, consultez la documentation du fabricant.

## Applications

Les applications s'exécutent sur l'appliance AWS Panorama pour effectuer des tâches de vision par ordinateur sur des flux vidéo. Vous pouvez créer des applications de vision par ordinateur en combinant du code Python et des modèles d'apprentissage automatique, puis les déployer sur l'appliance AWS Panorama via Internet. Les applications peuvent envoyer des vidéos sur un écran ou utiliser le SDK AWS pour envoyer les résultats aux services AWS.

Pour créer et déployer des applications, vous utilisez l'interface de ligne de commande d'application AWS Panorama. L'AWS Panorama Application CLI est un outil de ligne de commande qui génère des dossiers d'applications et des fichiers de configuration par défaut, crée des conteneurs avec Docker et télécharge des actifs. Vous pouvez exécuter plusieurs applications sur un seul appareil.

Pour plus d'informations, veuillez consulter Gestion d'AWS Panoramaapplications.

## Nœuds

Une application comprend plusieurs composants appelésnœuds, qui représentent les entrées, les sorties, les modèles et le code. Un nœud peut être configuré uniquement (entrées et sorties) ou inclure des artefacts (modèles et code). Les nœuds de code d'une application sont regroupés danspackages de nœudsque vous téléchargez sur un point d'accès Amazon S3, où l'appliance AWS Panorama peut y accéder. Unmanifeste de l'applicationest un fichier de configuration qui définit les connexions entre les nœuds.

Pour plus d'informations, veuillez consulter nœuds d'application.

## Modèles

Un modèle de vision par ordinateur est un réseau d'apprentissage automatique formé pour traiter des images. Les modèles de vision par ordinateur peuvent effectuer diverses tâches telles que la

classification, la détection, la segmentation et le suivi. Un modèle de vision par ordinateur prend une image en entrée et produit des informations sur l'image ou les objets de l'image.

AWS Panorama prend en charge les modèles créés avec PyTorch, Apache MXNet, et TensorFlow. Vous pouvez créer des modèles avec Amazon SageMaker ou dans votre environnement de développement. Pour plus d'informations, veuillez consulter ???.

# Configuration de l'appliance AWS Panorama

Pour commencer à utiliser votre appareil AWS Panorama ou un appareil compatible, enregistrez-le dans la console AWS Panorama et mettez à jour son logiciel. Au cours du processus de configuration, vous créez une ressource d'appliance dans AWS Panorama qui représente l'appliance physique et vous copiez des fichiers sur l'appliance à l'aide d'une clé USB. L'appliance utilise ces certificats et fichiers de configuration pour se connecter au service AWS Panorama. Vous utilisez ensuite la console AWS Panorama pour mettre à jour le logiciel de l'appliance et enregistrer les caméras.

Sections

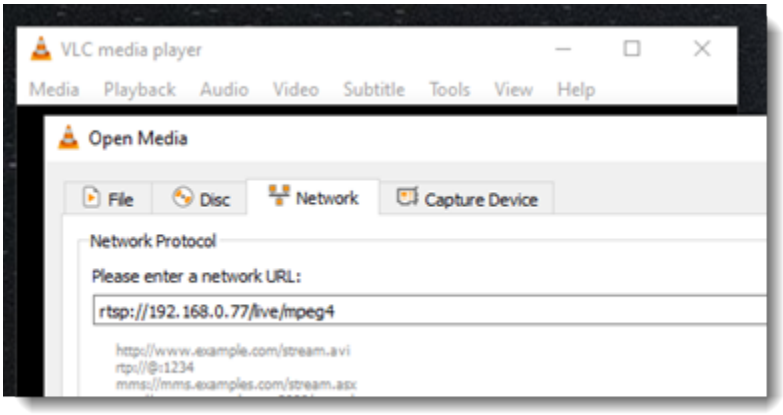


## Prérequis

Pour suivre ce didacticiel, vous avez besoin d'une version de travail AWS ou d'un appareil compatible et du matériel suivant :

- Affichage : écran doté d'une entrée HDMI permettant de visualiser la sortie de l'exemple d'application.
- Clé USB (incluse avec AWS Panorama Appliance) : clé USB 3.0 formatée en FAT32 avec au moins 1 Go de stockage, permettant de transférer une archive contenant des fichiers de configuration et un certificat vers l'appliance AWS Panorama.
- Caméra : caméra IP qui émet un flux vidéo RTSP.

Utilisez les outils et les instructions fournis par le fabricant de votre caméra pour identifier l'adresse IP et le chemin du flux de diffusion de la caméra. Vous pouvez utiliser un lecteur vidéo tel que VLC pour vérifier l'URL du flux en l'ouvrant en tant que source multimédia réseau :




La console AWS Panorama utilise d'autres services AWS pour assembler les composants de l'application, gérer les autorisations et vérifier les paramètres. Pour enregistrer une appliance et déployer l'exemple d'application, vous devez disposer des autorisations suivantes :

- AWSPanoramaFullAccess— Fournit un accès complet à AWS Panorama, aux points d'accès AWS Panorama dans Amazon S3, aux informations d'identification des appliances et aux journaux des appliances dans AmazonCloudWatch. AWS Secrets Manager Inclut l'autorisation de créer un rôle lié à un service pour AWS Panorama.
- AWS Identity and Access Management(IAM) : lors de la première exécution, pour créer des rôles utilisés par le service AWS Panorama et l'appliance AWS Panorama.

Si vous n'êtes pas autorisé à créer des rôles dans IAM, demandez à un administrateur d'ouvrir la console AWS Panorama et d'accepter l'invite de création de rôles de service.

## Enregistrez et configurez l'appliance AWS Panorama

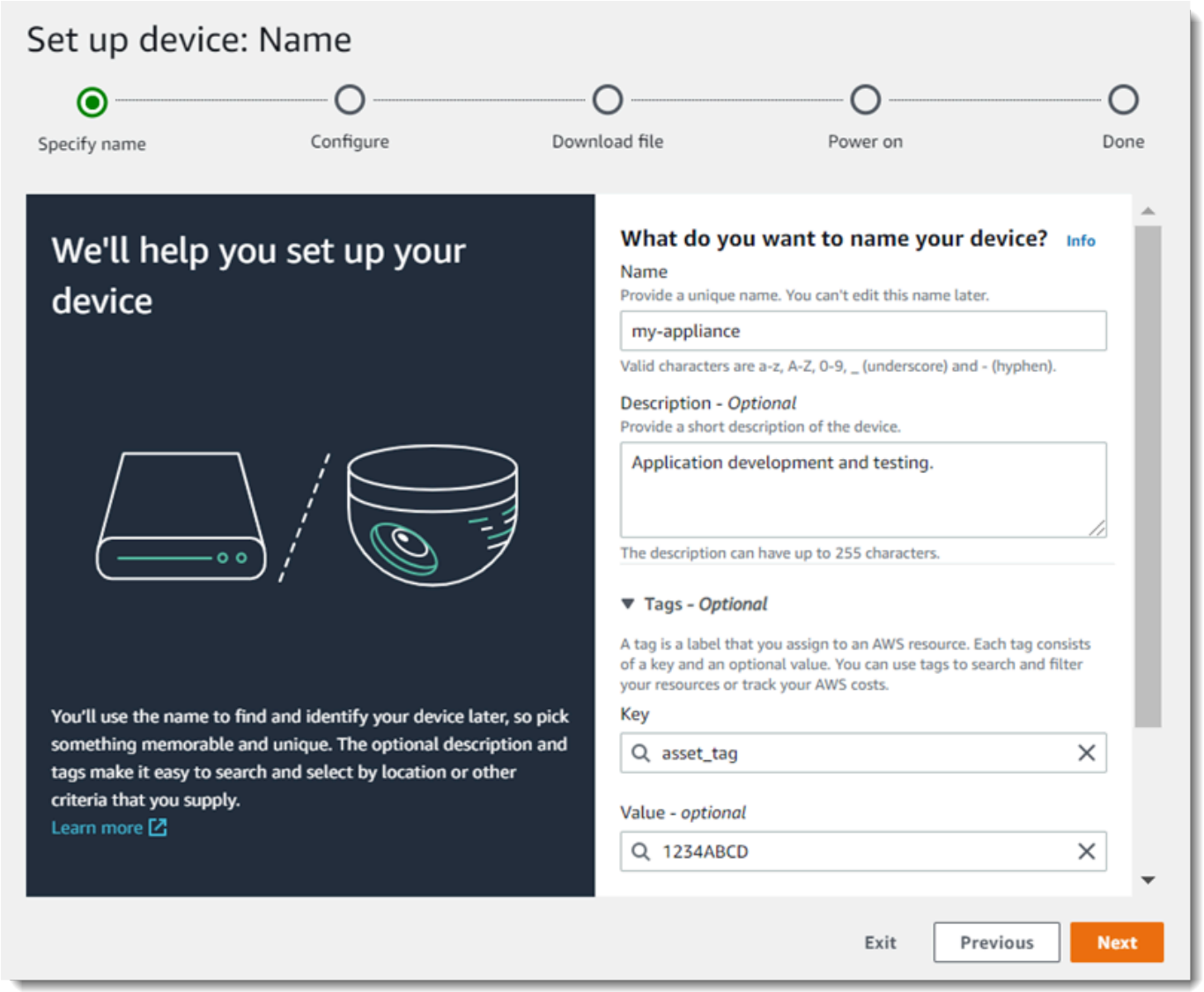
L'appliance AWS Panorama est un appareil matériel qui se connecte à des caméras connectées au réseau via une connexion réseau locale. Il utilise un système d'exploitation basé sur Linux qui inclut le SDK de l'application AWS Panorama et les logiciels de support permettant d'exécuter des applications de vision par ordinateur.

Pour se connecter à des AWS fins de gestion et de déploiement d'applications, l'appliance utilise un certificat d'appareil. Vous utilisez la console AWS Panorama pour générer un certificat de provisionnement. L'appliance utilise ce certificat temporaire pour terminer la configuration initiale et télécharger un certificat d'appareil permanent.

> **Important**
>
> Le certificat d'approvisionnement que vous générez au cours de cette procédure n'est valide que pendant 5 minutes. Si vous ne terminez pas le processus d'inscription dans ce délai, vous devez recommencer.

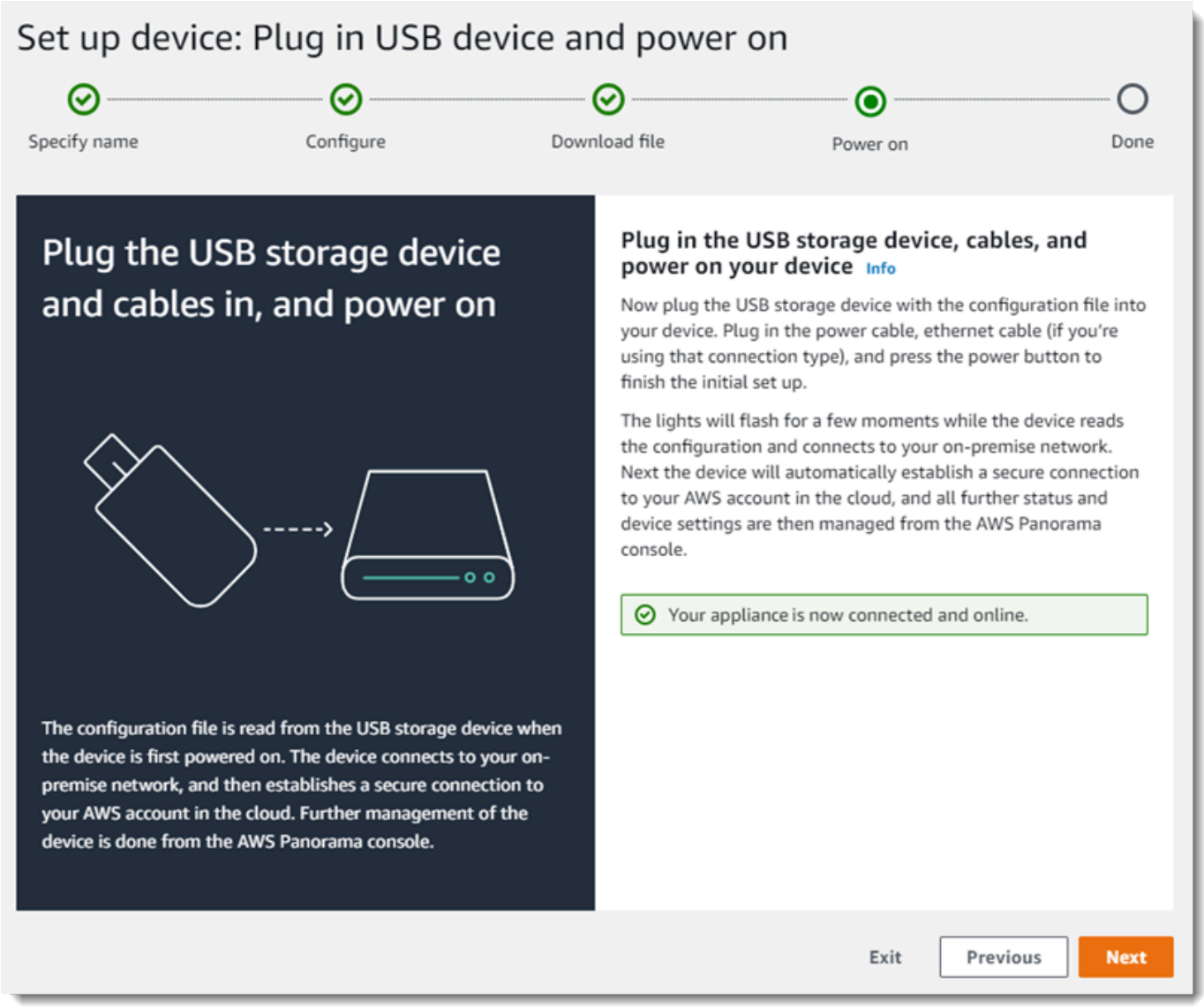
Pour enregistrer un appareil

1. Connect la clé USB à votre ordinateur. Préparez l'appliance en connectant les câbles réseau et d'alimentation. L'appliance s'allume et attend qu'une clé USB soit connectée.
2. Ouvrez la page de démarrage de la console AWS Panorama.
3. Choisissez Ajouter un appareil.
4. Choisissez Commencer la configuration.
5. Saisissez un nom et une description pour la ressource de l'appareil qui représente l'appliance dans AWS Panorama. Choisissez Next (Suivant)




6. Si vous devez attribuer manuellement une adresse IP, un serveur NTP ou des paramètres DNS, choisissez Paramètres réseau avancés. Sinon, choisissez Next (Suivant).
7. Choisissez Télécharger l'archive. Choisissez Suivant.
8. Copiez l'archive de configuration dans le répertoire racine de la clé USB.
9. Connect la clé USB au port USB 3.0 situé à l'avant de l'appareil, à côté du port HDMI.

   Lorsque vous connectez la clé USB, l'appliance copie l'archive de configuration et le fichier de configuration réseau sur elle-même et se connecte au AWS Cloud. Le voyant d'état de l'appliance passe du vert au bleu lorsque la connexion est terminée, puis revient au vert.
10. Pour continuer, choisissez Suivant.



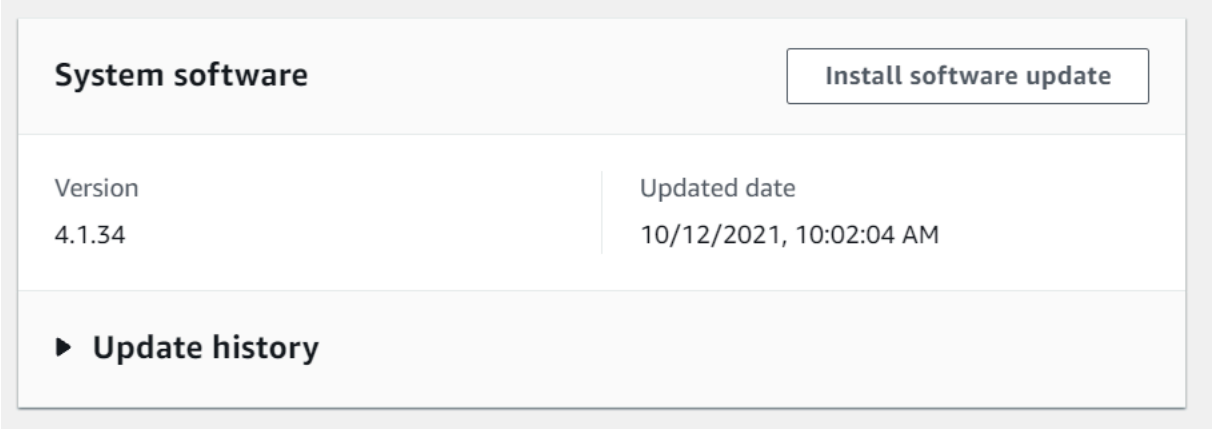

11. Sélectionnez Done (Exécuté).

## Mettre à niveau le logiciel de l'appliance

L'appliance AWS Panorama comporte plusieurs composants logiciels, notamment un système d'exploitation Linux, le SDK de l'application AWS Panorama, ainsi que des bibliothèques et des frameworks de vision par ordinateur compatibles. Pour vous assurer de pouvoir utiliser les fonctionnalités et applications les plus récentes avec votre appliance, mettez à niveau son logiciel après l'installation et chaque fois qu'une mise à jour est disponible.

Pour mettre à jour le logiciel de l'appliance

1. Ouvrez la page Appareils de la console AWS Panorama.
2. Choisissez un appareil.
3. Choisissez Réglages
4. Sous Logiciel système, choisissez Installer la mise à jour logicielle.




5. Choisissez une nouvelle version, puis choisissez Installer.

> **Important**
>
> Avant de continuer, retirez la clé USB de l'appliance et formatez-la pour supprimer son contenu. L'archive de configuration contient des données sensibles et n'est pas supprimée automatiquement.

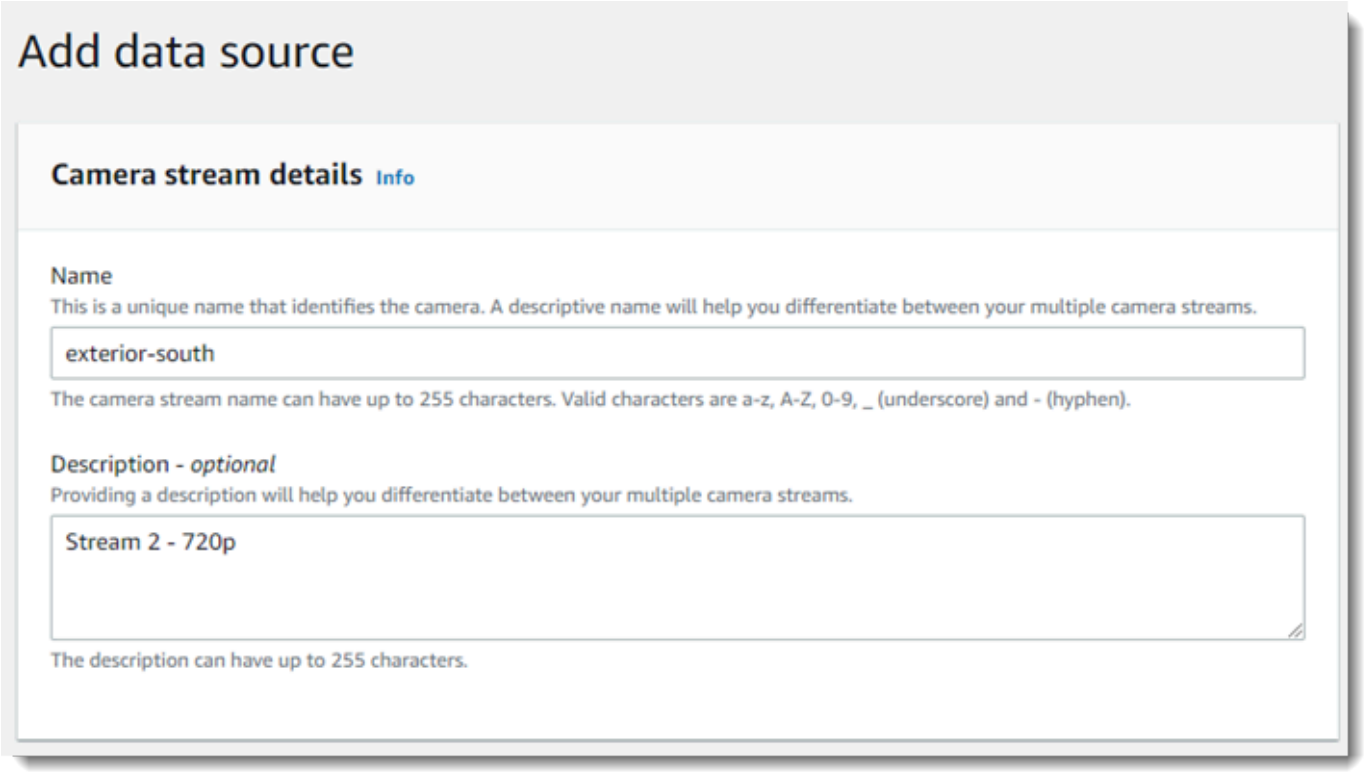
Le processus de mise à niveau peut prendre 30 minutes ou plus. Vous pouvez suivre sa progression dans la console AWS Panorama ou sur un moniteur connecté. Une fois le processus terminé, l'appliance redémarre.

## Ajouter un flux de caméra

Enregistrez ensuite un flux de caméra avec la console AWS Panorama.

Pour enregistrer un flux de caméra

1. Ouvrez la page Sources de données de la console AWS Panorama.
2. Choisissez Ajouter une source de données.




3. Configurez les paramètres suivants.

   - Nom : nom du flux de caméra.
   - Description : brève description de la caméra, de son emplacement ou d'autres détails.
   - URL RTSP : URL qui spécifie l'adresse IP de la caméra et le chemin d'accès au flux. Par exemple, `rtsp://192.168.0.77/live/mpeg4/`
   - Informations d'identification : si le flux de la caméra est protégé par un mot de passe, spécifiez le nom d'utilisateur et le mot de passe.
4. Choisissez Save (Enregistrer).

AWS Panorama stocke les informations d'identification de votre caméra en toute sécuritéAWS Secrets Manager. Plusieurs applications peuvent traiter simultanément le même flux de caméra.

## Étapes suivantes

Si vous avez rencontré des erreurs lors de l'installation, consultezRésolution des problèmes.

Pour déployer un exemple d'application, passez à la rubrique suivante.

# Déploiement du modèle d'application AWS Panorama

Une fois que vous avezconfigurer votre appliance AWS Panorama ou un appareil compatibleet a mis à jour son logiciel, déployé un exemple d'application. Dans les sections suivantes, vous devez importer un exemple d'application avec l'interface de ligne de commande de l'application AWS Panorama et le déployer avec la console AWS Panorama.

L'exemple d'application utilise un modèle d'apprentissage automatique pour classer des objets dans des images vidéo provenant d'une caméra réseau. Il utilise le SDK de l'application AWS Panorama pour charger un modèle, obtenir des images et exécuter le modèle. L'application superpose ensuite les résultats au-dessus de la vidéo d'origine et les transmet sur un écran connecté.

Dans un environnement de vente au détail, l'analyse des modèles de trafic piétonnier permet de prévoir les niveaux de trafic. En combinant l'analyse avec d'autres données, vous pouvez planifier l'augmentation des besoins en personnel pendant les vacances et autres événements, mesurer l'efficacité des publicités et des promotions commerciales ou optimiser l'emplacement des présentoirs et la gestion des stocks.

Sections



## Prérequis

Pour suivre les procédures décrites dans ce didacticiel, vous aurez besoin d'un shell ou d'un terminal de ligne de commande pour exécuter des commandes. Dans les listes de code, les commandes sont précédées d'un symbole d'invite ($) et du nom du répertoire actuel, le cas échéant.

```
~/panorama-project$ this is a command
this is output
```

Pour les commandes longues, nous utilisons un caractère d'échappement (\) pour répartir une commande sur plusieurs lignes.

Sur Linux et macOS, utilisez votre gestionnaire de shell et de package préféré. Sur Windows 10, vous pouvez installer le sous-système Windows pour Linux afin d'obtenir une version intégrée à Windows d'Ubuntu et Bash. Pour obtenir de l'aide sur la configuration d'un environnement de développement sous Windows, voirConfiguration d'un environnement de développement sous Windows.

Vous utilisez Python pour développer des applications AWS Panorama et installer des outils avec pip, le gestionnaire de packages de Python. Si vous n'avez pas déjà Python,installer la dernière version. Si vous utilisez Python 3 mais pas pip, installez pip avec le gestionnaire de packages de votre système d'exploitation ou installez une nouvelle version de Python, fournie avec pip.

Dans ce didacticiel, vous allez utiliser Docker pour créer le conteneur qui exécute le code de votre application. Installez Docker depuis le site Web de Docker : Obtenez Docker

Ce didacticiel utilise l'interface de ligne de commande de l'application AWS Panorama pour importer l'exemple d'application, créer des packages et télécharger des artefacts. La CLI de l'application AWS Panorama utiliseAWS Command Line Interface(AWS CLI) pour appeler les opérations d'API de service. Si vous avez déjàAWS CLI, mettez-le à jour vers la dernière version. Pour installer la CLI de l'application AWS Panorama etAWS CLI, utilisation`pip`.

```
$ pip3 install --upgrade awscli panoramacli
```

Téléchargez l'exemple d'application et extrayez-le dans votre espace de travail.

- Exemple d'application–aws-panorama-sample.zip

## Importer l'exemple d'application

Pour importer l'exemple d'application à utiliser dans votre compte, utilisez l'interface de ligne de commande de l'application AWS Panorama. Les dossiers et le manifeste de l'application contiennent des références à un numéro de compte fictif. Pour les mettre à jour avec votre numéro de compte, exécutez le`panorama-cli import-application`commande.

```
aws-panorama-sample$ panorama-cli import-application
```

Dans laSAMPLE_CODEpackage, dans lepackagesrépertoire, contient le code et la configuration de l'application, y compris un Dockerfile qui utilise l'image de base de l'application,`panorama-application`. Pour créer le conteneur d'applications qui s'exécute sur l'appliance, utilisez`panorama-cli build-container`commande.

```
aws-panorama-sample$ ACCOUNT_ID=$(aws sts get-caller-identity --output text --query
 'Account')
aws-panorama-sample$ panorama-cli build-container --container-asset-name code_asset --
package-path packages/${ACCOUNT_ID}-SAMPLE_CODE-1.0
```

La dernière étape avec l'interface de ligne de commande de l'application AWS Panorama consiste à enregistrer le code et les nœuds de modèle de l'application, et à charger les actifs vers un point d'accès Amazon S3 fourni par le service. Les ressources incluent l'image du conteneur du code, le modèle et un fichier descripteur pour chacun d'entre eux. Pour enregistrer les nœuds et télécharger des actifs, exécutez le`panorama-cli package-application`commande.

```
aws-panorama-sample$ panorama-cli package-application
Uploading package model
Registered model with patch version
 bc9c58bd6f83743f26aa347dc86bfc3dd2451b18f964a6de2cc4570cb6f891f9
Uploading package code
Registered code with patch version
 11fd7001cb31ea63df6aaed297d600a5ecf641a987044a0c273c78ceb3d5d806
```

## Déployer l'application

Utilisez la console AWS Panorama pour déployer l'application sur votre appliance.

Pour déployer l'application

1. Ouvrez la console AWS PanoramaPage Applications déployées.
2. ChoisissezDéploiement.
3. Collez le contenu du manifeste de l'application,`graphs/aws-panorama-sample/graph.json`, dans l'éditeur de texte. Choisissez Next (Suivant).
4. Dans Application name (Nom de l'application), saisissez `aws-panorama-sample`.
5. ChoisissezProcéder au déploiement.
6. ChoisissezCommencer le déploiement.

7. ChoisissezSuivantsans sélectionner de rôle.
8. ChoisissezSélectionnez un appareil, puis choisissez votre appareil. Choisissez Next (Suivant).
9. Sur leSélectionnez les sources de donnéesétape, choisissezAfficher les entrées, et ajoutez le flux de votre caméra en tant que source de données. Choisissez Next (Suivant).
10. Sur leConfigurationétape, choisissezSuivant.
11. ChoisissezDéploiement, puisTerminé.
12. Dans la liste des applications déployées, choisissezaws-panorama-sample.

Actualisez cette page pour obtenir des mises à jour ou utilisez le script suivant pour surveiller le déploiement depuis la ligne de commande.

Example monitor-deployment.sh

```
while true; do
  aws panorama list-application-instances --query 'ApplicationInstances[?Name==`aws-
panorama-sample`]'
  sleep 10
done
```

```
[
    {
        "Name": "aws-panorama-sample",
        "ApplicationInstanceId": "applicationInstance-x264exmpl33gq5pchc2ekoi6uu",
        "DefaultRuntimeContextDeviceName": "my-appliance",
        "Status": "DEPLOYMENT_PENDING",
        "HealthStatus": "NOT_AVAILABLE",
        "StatusDescription": "Deployment Workflow has been scheduled.",
        "CreatedTime": 1630010747.443,
        "Arn": "arn:aws:panorama:us-west-2:123456789012:applicationInstance/
applicationInstance-x264exmpl33gq5pchc2ekoi6uu",
        "Tags": {}
    }
]
[
    {
        "Name": "aws-panorama-sample",
        "ApplicationInstanceId": "applicationInstance-x264exmpl33gq5pchc2ekoi6uu",
        "DefaultRuntimeContextDeviceName": "my-appliance",
```

```
        "Status": "DEPLOYMENT_PENDING",
        "HealthStatus": "NOT_AVAILABLE",
        "StatusDescription": "Deployment Workflow has completed data validation.",
        "CreatedTime": 1630010747.443,
        "Arn": "arn:aws:panorama:us-west-2:123456789012:applicationInstance/
applicationInstance-x264exmpl33gq5pchc2ekoi6uu",
        "Tags": {}
    }
]
...
```

Si l'application ne démarre pas, vérifiez lejournaux des applications et des appareilsdans Amazon CloudWatch Journaux.

## Afficher la sortie

Une fois le déploiement terminé, l'application commence à traiter le flux vidéo et envoie des journaux à CloudWatch.

Pour consulter les connexions CloudWatch Journaux

1. Ouvrez lePage Groupes de journaux du CloudWatch Console Logs.
2. Trouvez les journaux de l'application et de l'appliance AWS Panorama dans les groupes suivants :

   - Journaux des appareils–`/aws/panorama/devices/`*`device-id`*
   - Journaux d'applications–`/aws/panorama/devices/`*`device-id`*`/applications/`*`instance-id`*

```
2022-08-26 17:43:39 INFO     INITIALIZING APPLICATION
2022-08-26 17:43:39 INFO     ## ENVIRONMENT VARIABLES
{'PATH': '/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin', 'TERM':
 'xterm', 'container': 'podman'...}
2022-08-26 17:43:39 INFO     Configuring parameters.
2022-08-26 17:43:39 INFO     Configuring AWS SDK for Python.
2022-08-26 17:43:39 INFO     Initialization complete.
2022-08-26 17:43:39 INFO     PROCESSING STREAMS
2022-08-26 17:46:19 INFO     epoch length: 160.183 s (0.936 FPS)
2022-08-26 17:46:19 INFO     avg inference time: 805.597 ms
2022-08-26 17:46:19 INFO     max inference time: 120023.984 ms
```

```
2022-08-26 17:46:19 INFO     avg frame processing time: 1065.129 ms
2022-08-26 17:46:19 INFO     max frame processing time: 149813.972 ms
2022-08-26 17:46:29 INFO     epoch length: 10.562 s (14.202 FPS)
2022-08-26 17:46:29 INFO     avg inference time: 7.185 ms
2022-08-26 17:46:29 INFO     max inference time: 15.693 ms
2022-08-26 17:46:29 INFO     avg frame processing time: 66.561 ms
2022-08-26 17:46:29 INFO     max frame processing time: 123.774 ms
```

Pour visualiser la sortie vidéo de l'application, connectez l'appareil à un moniteur à l'aide d'un câble HDMI. Par défaut, l'application affiche tout résultat de classification présentant un niveau de confiance supérieur à 20 %.

Example squeezenet_classes.json

```
["tench", "goldfish", "great white shark", "tiger shark",
"hammerhead", "electric ray", "stingray", "cock", "hen", "ostrich",
"brambling", "goldfinch", "house finch", "junco", "indigo bunting",
"robin", "bulbul", "jay", "magpie", "chickadee", "water ouzel",
"kite", "bald eagle", "vulture", "great grey owl",
"European fire salamander", "common newt", "eft",
"spotted salamander", "axolotl", "bullfrog", "tree frog",
...
```

Le modèle d'échantillon comprend 1 000 classes, y compris de nombreux animaux, de la nourriture et des objets courants. Essayez de pointer votre appareil photo vers un clavier ou une tasse à café.


Pour des raisons de simplicité, l'exemple d'application utilise un modèle de classification léger. Le modèle produit un tableau unique avec une probabilité pour chacune de ses classes. Les applications du monde réel utilisent plus fréquemment des modèles de détection d'objets dotés d'une sortie multidimensionnelle. Pour des exemples d'applications avec des modèles plus complexes, voirExemples d'applications, de scripts et de modèles.

## Activer le kit SDK pour Python

L'exemple d'application utilise l'AWS SDK for Python (Boto)pour envoyer les métriques à Amazon CloudWatch. Pour activer cette fonctionnalité, créez un rôle qui autorise l'application à envoyer des métriques et redéployez l'application avec le rôle associé.

L'exemple d'application inclut unAWS CloudFormationmodèle qui crée un rôle avec les autorisations dont il a besoin. Pour créer le rôle, utilisez l'`aws cloudformation deploy`commande.

```
$ aws cloudformation deploy --template-file aws-panorama-sample.yml --stack-name aws-panorama-sample-runtime --capabilities CAPABILITY_NAMED_IAM
```

Pour redéployer l'application

1. Ouvrez la console AWS PanoramaPage Applications déployées.
2. Choisissez une application.
3. Choisissez Remplacer.
4. Suivez les étapes pour déployer l'application. Dans leSpécification du rôle IAM, choisissez le rôle que vous avez créé. Son nom commence par `aws-panorama-sample-runtime`.
5. Lorsque le déploiement est terminé, ouvrez leCloudWatchconsoleet consultez les métriques dans le`AWSPanoramaApplication`Espaces de noms. Toutes les 150 images, l'application enregistre et télécharge des métriques pour le traitement des images et le temps d'inférence.

## Nettoyage

Si vous avez fini de travailler avec l'exemple d'application, vous pouvez utiliser la console AWS Panorama pour le supprimer de l'appliance.

Pour supprimer l'application de l'appliance

1. Ouvrez la console AWS PanoramaPage Applications déployées.
2. Choisissez une application.
3. ChoisissezSuppression de l'appareil.

## Étapes suivantes

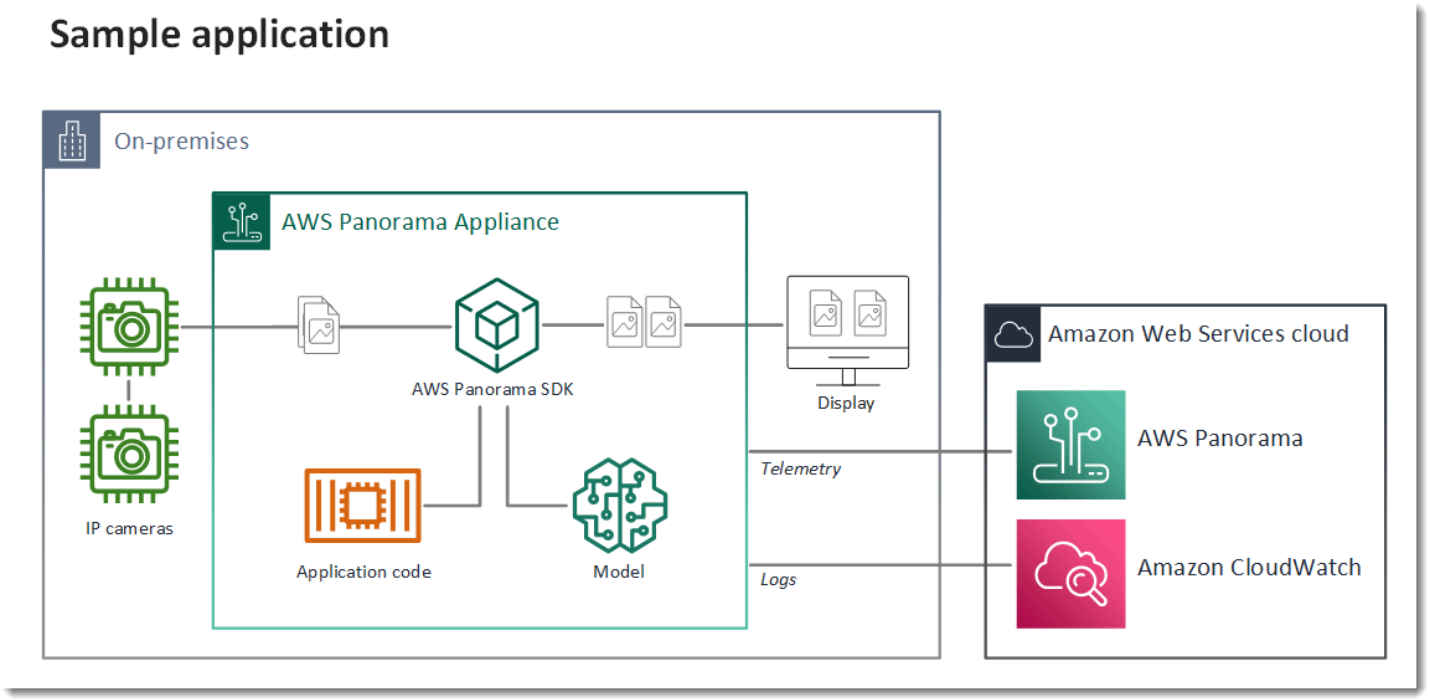
Si vous avez rencontré des erreurs lors du déploiement ou de l'exécution de l'exemple d'application, voirRésolution des problèmes.

Pour en savoir plus sur les fonctionnalités et la mise en œuvre de l'exemple d'application, passez àle sujet suivant.

# Développement d'applications AWS Panorama

Vous pouvez utiliser l'exemple d'application pour en savoir plus sur la structure d'application AWS Panorama et comme point de départ pour votre propre application.

Le schéma suivant illustre les principaux composants de l'application exécutée sur une appliance AWS Panorama. Le code de l'application utilise le kit SDK de l'application AWS Panorama pour obtenir des images et interagir avec le modèle, auquel il n'a pas d'accès direct. L'application envoie la vidéo vers un écran connecté mais n'envoie pas de données d'image en dehors de votre réseau local.




Dans cet exemple, l'application utilise le kit SDK d'application AWS Panorama pour obtenir des images vidéo d'une caméra, prétraiter les données vidéo et envoyer les données à un modèle de vision artificielle qui détecte les objets. L'application affiche le résultat sur un écran HDMI connecté à l'appareil.

Sections

- Chargement de métriques avec le kit SDK pour Python
- Étapes suivantes

## Le manifeste d'application

Le manifeste de l'application est un fichier nommégraph.jsondans legraphsfolder. Le manifeste définit les composants de l'application, à savoir les packages, les nœuds et les bords.

Les packages sont des fichiers de code, de configuration et binaires pour le code d'application, les modèles, les caméras et les affichages. L'exemple d'application utilise 4 packages :

Example **graphs/aws-panorama-sample/graph.json**— Packages

```
    "packages": [
        {
            "name": "123456789012::SAMPLE_CODE",
            "version": "1.0"
        },
        {
            "name": "123456789012::SQUEEZENET_PYTORCH_V1",
            "version": "1.0"
        },
        {
            "name": "panorama::abstract_rtsp_media_source",
            "version": "1.0"
        },
        {
            "name": "panorama::hdmi_data_sink",
            "version": "1.0"
        }
    ],
```

Les deux premiers packages sont définis dans l'application, dans lepackagesannuaire. Ils contiennent le code et le modèle spécifiques à cette application. Les deux autres packages sont des packages génériques de caméra et d'affichage fournis par le service AWS Panorama. Leabstract_rtsp_media_sourceest un espace réservé pour une caméra que vous remplacez lors du déploiement. Lehdmi_data_sinkreprésente le connecteur de sortie HDMI de l'appareil.

Les nœuds sont des interfaces vers des packages, ainsi que des paramètres autres que des packages qui peuvent avoir des valeurs par défaut que vous remplacez au moment du déploiement. Les packages de code et de modèle définissent les interfaces danspackage.jsondes fichiers qui

spécifient des entrées et des sorties, qui peuvent être des flux vidéo ou un type de données de base tel qu'un flottant, un booléen ou une chaîne.

Par exemple, les recettes`code_node`nœud fait référence à une interface du`SAMPLE_CODE`package.

```
        "nodes": [
            {
                "name": "code_node",
                "interface": "123456789012::SAMPLE_CODE.interface",
                "overridable": false,
                "launch": "onAppStart"
            },
```

Cette interface est définie dans le fichier de configuration de package,`package.json`. L'interface indique que le package est une logique métier et qu'il prend un flux vidéo nommé`video_in`et un nombre à virgule flottante nommé`threshold`en tant qu'entrées. L'interface indique également que le code nécessite un tampon de flux vidéo nommé`video_out`pour sortir une vidéo sur un écran

Example **`packages/123456789012-SAMPLE_CODE-1.0/package.json`**

```
{
    "nodePackage": {
        "envelopeVersion": "2021-01-01",
        "name": "SAMPLE_CODE",
        "version": "1.0",
        "description": "Computer vision application code.",
        "assets": [],
        "interfaces": [
            {
                "name": "interface",
                "category": "business_logic",
                "asset": "code_asset",
                "inputs": [
                    {
                        "name": "video_in",
                        "type": "media"
                    },
                    {
                        "name": "threshold",
                        "type": "float32"
                    }
                ],
                "outputs": [
```

```
                    {
                        "description": "Video stream output",
                        "name": "video_out",
                        "type": "media"
                    }
                ]
            }
        ]
    }
}
```

De retour dans le manifeste de l'application, le`camera_node`node représente un flux vidéo provenant d'une caméra. Il inclut un décorateur qui apparaît dans la console lorsque vous déployez l'application, vous invitant à choisir un flux de caméra.

Example **`graphs/aws-panorama-sample/graph.json`**— Nœud caméra

```
            {
                "name": "camera_node",
                "interface": "panorama::abstract_rtsp_media_source.rtsp_v1_interface",
                "overridable": true,
                "launch": "onAppStart",
                "decorator": {
                    "title": "Camera",
                    "description": "Choose a camera stream."
                }
            },
```

Un nœud de paramètre,`threshold_param`, définit le paramètre de seuil de confiance utilisé par le code d'application. Elle a une valeur par défaut de 60 et peut être remplacée pendant le déploiement.

Example **`graphs/aws-panorama-sample/graph.json`**— Nœud paramètre

```
            {
                "name": "threshold_param",
                "interface": "float32",
                "value": 60.0,
                "overridable": true,
                "decorator": {
                    "title": "Confidence threshold",
                    "description": "The minimum confidence for a classification to be
 recorded."
                }
```

```
        }
```

La dernière section du manifeste de candidature,`edges`, établit des connexions entre les nœuds. Le flux vidéo de la caméra et le paramètre de seuil se connectent à l'entrée du nœud de code, et la sortie vidéo du nœud de code se connecte à l'écran.

Example **`graphs/aws-panorama-sample/graph.json`**— Bords

```
    "edges": [
        {
            "producer": "camera_node.video_out",
            "consumer": "code_node.video_in"
        },
        {
            "producer": "code_node.video_out",
            "consumer": "output_node.video_in"
        },
        {
            "producer": "threshold_param",
            "consumer": "code_node.threshold"
        }
    ]
```

## Create avec l'exemple d'application

Vous pouvez utiliser l'exemple d'application comme point de départ pour votre propre application.

Le nom de chaque package doit être unique dans votre compte. Si vous et un autre utilisateur de votre compte utilisez tous deux un nom de package générique tel que`codeoumodel`, il se peut que vous obteniez la mauvaise version du package lors du déploiement. Remplacez le nom du package de code par un nom qui représente votre application.

Pour renommer le package de code

1. Renommez le dossier du package :`packages/123456789012-`*`SAMPLE_CODE`*`-1.0/`.
2. Mettez à jour le nom du package dans les emplacements suivants.

   - Le manifeste d'application–`graphs/aws-panorama-sample/graph.json`
   - Configuration de package–`packages/123456789012-SAMPLE_CODE-1.0/package.json`
   - Script de build–`3-build-container.sh`

Pour mettre à jour le code de l'application

1. Modifiez le code de l'application dans`packages/123456789012-SAMPLE_CODE-1.0/src/application.py`.
2. Pour générer le conteneur, exécutez`3-build-container.sh`.

```
aws-panorama-sample$ ./3-build-container.sh
TMPDIR=$(pwd) docker build -t code_asset packages/123456789012-SAMPLE_CODE-1.0
Sending build context to Docker daemon  61.44kB
Step 1/2 : FROM public.ecr.aws/panorama/panorama-application
 ---> 9b197f256b48
Step 2/2 : COPY src /panorama
 ---> 55c35755e9d2
Successfully built 55c35755e9d2
Successfully tagged code_asset:latest
docker export --output=code_asset.tar $(docker create code_asset:latest)
gzip -9 code_asset.tar
Updating an existing asset with the same name
{
    "name": "code_asset",
    "implementations": [
        {
            "type": "container",
            "assetUri":
 "98aaxmpl1c1ef64cde5ac13bd3be5394e5d17064beccee963b4095d83083c343.tar.gz",
            "descriptorUri":
 "1872xmpl129481ed053c52e66d6af8b030f9eb69b1168a29012f01c7034d7a8f.json"
        }
    ]
}
Container asset for the package has been succesfully built at  ~/aws-panorama-
sample-dev/
assets/98aaxmpl1c1ef64cde5ac13bd3be5394e5d17064beccee963b4095d83083c343.tar.gz
```

   L'interface de ligne de commande supprime automatiquement l'ancien actif de conteneur du`assets`dossier et met à jour la configuration du package.
3. Pour charger les packages, exécutez`4-package-application.py`.
4. Ouvrez la console AWS PanoramaPage Applications déployées.
5. Choisissez une application.
6. Choisissez Remplacer.

7. Procédez comme suit pour déployer l'application. Si nécessaire, vous pouvez modifier le manifeste de l'application, les flux de caméra ou les paramètres.

## Changer le modèle de vision par ordinateur

L'exemple d'application inclut un modèle de vision par ordinateur. Pour utiliser votre propre modèle, modifiez la configuration du nœud du modèle et utilisez l'interface de ligne de commande de l'application AWS Panorama pour l'importer en tant que ressource.

L'exemple suivant utilise un SSD MXNet ResNet50 modèles que vous pouvez télécharger à partir de ce guide GitHub repo :ssd_512_resnet50_v1_voc.tar.gz

Pour modifier le modèle de l'exemple d'application

1. Renommez le dossier du package en fonction de votre modèle. Par exemple, pour`packages/`*`123456789012`*`-`*`SSD_512_RESNET50_V1_VOC`*`-1.0/`.
2. Mettez à jour le nom du package dans les emplacements suivants.

   - Le manifeste d'application–`graphs/aws-panorama-sample/graph.json`
   - Configuration de package–`packages/`*`123456789012`*`-`*`SSD_512_RESNET50_V1_VOC`*`-1.0/package.json`
3. Dans le fichier de configuration du package (`package.json`). Remplacez le`assets`valeur dans un tableau vide.

   ```
   {
       "nodePackage": {
           "envelopeVersion": "2021-01-01",
           "name": "SSD_512_RESNET50_V1_VOC",
           "version": "1.0",
           "description": "Compact classification model",
           "assets": [],
   ```
4. Ouvrez le fichier descripteur de package (`descriptor.json`). Mettre à jour le`framework`et`shape`valeurs correspondant à votre modèle.

   ```
   {
       "mlModelDescriptor": {
           "envelopeVersion": "2021-01-01",
   ```

```
        "framework": "MXNET",
        "inputs": [
            {
                "name": "data",
                "shape": [ 1, 3, 512, 512 ]
            }
        ]
    }
}
```

La valeur pourforme,`1,3,512,512`, indique le nombre d'images que le modèle prend en entrée (1), le nombre de canaux dans chaque image (3 : rouge, vert et bleu) et les dimensions de l'image (512 x 512). Les valeurs et l'ordre du tableau varient selon les modèles.

5. Importez le modèle avec l'interface de ligne de commande d'application AWS Panorama. L'interface de ligne de commande de l'application AWS Panorama copie les fichiers de modèle et de descripteur dans le`assets`avec des noms uniques, et met à jour la configuration du package.

```
aws-panorama-sample$ panorama-cli add-raw-model --model-asset-name model-asset \
--model-local-path ssd_512_resnet50_v1_voc.tar.gz \
--descriptor-path packages/123456789012-SSD_512_RESNET50_V1_VOC-1.0/descriptor.json
 \
--packages-path packages/123456789012-SSD_512_RESNET50_V1_VOC-1.0
{
    "name": "model-asset",
    "implementations": [
        {
            "type": "model",
            "assetUri":
 "b1a1589afe449b346ff47375c284a1998c3e1522b418a7be8910414911784ce1.tar.gz",
            "descriptorUri":
 "a6a9508953f393f182f05f8beaa86b83325f4a535a5928580273e7fe26f79e78.json"
        }
    ]
}
```

6. Pour télécharger le modèle, exécutez`panorama-cli package-application`.

```
$ panorama-cli package-application
Uploading package SAMPLE_CODE
Patch Version 1844d5a59150d33f6054b04bac527a1771fd2365e05f990ccd8444a5ab775809
 already registered, ignoring upload
Uploading package SSD_512_RESNET50_V1_VOC
```

```
Patch version for the package
 244a63c74d01e082ad012ebf21e67eef5d81ce0de4d6ad1ae2b69d0bc498c8fd
upload: assets/
b1a1589afe449b346ff47375c284a1998c3e1522b418a7be8910414911784ce1.tar.gz to
 s3://arn:aws:s3:us-west-2:454554846382:accesspoint/panorama-123456789012-
wc66m5eishf4si4sz5jefhx
63a/123456789012/nodePackages/SSD_512_RESNET50_V1_VOC/binaries/
b1a1589afe449b346ff47375c284a1998c3e1522b418a7be8910414911784ce1.tar.gz
upload: assets/
a6a9508953f393f182f05f8beaa86b83325f4a535a5928580273e7fe26f79e78.json to
 s3://arn:aws:s3:us-west-2:454554846382:accesspoint/panorama-123456789012-
wc66m5eishf4si4sz5jefhx63
a/123456789012/nodePackages/SSD_512_RESNET50_V1_VOC/binaries/
a6a9508953f393f182f05f8beaa86b83325f4a535a5928580273e7fe26f79e78.json
{
    "ETag": "\"2381dabba34f4bc0100c478e67e9ab5e\"",
    "ServerSideEncryption": "AES256",
    "VersionId": "KbY5fpESdpYamjWZ0YyGqHo3.LQQWUC2"
}
Registered SSD_512_RESNET50_V1_VOC with patch version
 244a63c74d01e082ad012ebf21e67eef5d81ce0de4d6ad1ae2b69d0bc498c8fd
Uploading package SQUEEZENET_PYTORCH_V1
Patch Version 568138c430e0345061bb36f05a04a1458ac834cd6f93bf18fdacdffb62685530
 already registered, ignoring upload
```

7. Mettez à jour le code d'application. La plupart du code peut être réutilisé. Le code spécifique à la réponse du modèle se trouve dans le`process_results`Méthode.

```
    def process_results(self, inference_results, stream):
        """Processes output tensors from a computer vision model and annotates a
 video frame."""
        for class_tuple in inference_results:
            indexes = self.topk(class_tuple[0])
        for j in range(2):
            label = 'Class [%s], with probability %.3f.'%
 (self.classes[indexes[j]], class_tuple[0][indexes[j]])
            stream.add_label(label, 0.1, 0.25 + 0.1*j)
```

En fonction de votre modèle, vous devrez peut-être également mettre à jour`preprocess`Méthode.

## Traitement des images

Avant d'envoyer une image au modèle, l'application la prépare pour l'inférence en la redimensionnant et en normalisant les données de couleur. Le modèle utilisé par l'application nécessite une image de 224 x 224 pixels avec trois canaux de couleur, pour correspondre au nombre d'entrées de sa première couche. L'application ajuste chaque valeur de couleur en la convertissant en un nombre compris entre 0 et 1, en soustrayant la valeur moyenne de cette couleur et en la divisant par l'écart type. Enfin, il combine les canaux de couleur et les convertit en NumPy tableau que le modèle peut traiter.

Example application.py— Prétraitement

```
    def preprocess(self, img, width):
        resized = cv2.resize(img, (width, width))
        mean = [0.485, 0.456, 0.406]
        std = [0.229, 0.224, 0.225]
        img = resized.astype(np.float32) / 255.
        img_a = img[:, :, 0]
        img_b = img[:, :, 1]
        img_c = img[:, :, 2]
        # Normalize data in each channel
        img_a = (img_a - mean[0]) / std[0]
        img_b = (img_b - mean[1]) / std[1]
        img_c = (img_c - mean[2]) / std[2]
        # Put the channels back together
        x1 = [[[], [], []]]
        x1[0][0] = img_a
        x1[0][1] = img_b
        x1[0][2] = img_c
        return np.asarray(x1)
```

Ce processus donne les valeurs du modèle dans une plage prévisible centrée autour de 0. Il correspond au prétraitement appliqué aux images du jeu de données d'entraînement, qui est une approche standard mais qui peut varier selon le modèle.

## Chargement de métriques avec le kit SDK pour Python

L'exemple d'application utilise le SDK pour Python pour télécharger des métriques vers Amazon. CloudWatch.

Example application.py— SDK pour Python

```
    def process_streams(self):
        """Processes one frame of video from one or more video streams."""
        ...
            logger.info('epoch length: {:.3f} s ({:.3f} FPS)'.format(epoch_time,
epoch_fps))
            logger.info('avg inference time: {:.3f} ms'.format(avg_inference_time))
            logger.info('max inference time: {:.3f} ms'.format(max_inference_time))
            logger.info('avg frame processing time: {:.3f}
ms'.format(avg_frame_processing_time))
            logger.info('max frame processing time: {:.3f}
ms'.format(max_frame_processing_time))
            self.inference_time_ms = 0
            self.inference_time_max = 0
            self.frame_time_ms = 0
            self.frame_time_max = 0
            self.epoch_start = time.time()
            self.put_metric_data('AverageInferenceTime', avg_inference_time)
            self.put_metric_data('AverageFrameProcessingTime',
avg_frame_processing_time)

    def put_metric_data(self, metric_name, metric_value):
        """Sends a performance metric to CloudWatch."""
        namespace = 'AWSPanoramaApplication'
        dimension_name = 'Application Name'
        dimension_value = 'aws-panorama-sample'
        try:
            metric = self.cloudwatch.Metric(namespace, metric_name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{
                    'MetricName': metric_name,
                    'Value': metric_value,
                    'Unit': 'Milliseconds',
                    'Dimensions': [
                        {
                            'Name': dimension_name,
                            'Value': dimension_value
                        },
                        {
                            'Name': 'Device ID',
                            'Value': self.device_id
                        }
```

```
                    ]
                }]
            )
            logger.info("Put data for metric %s.%s", namespace, metric_name)
        except ClientError:
            logger.warning("Couldn't put data for metric %s.%s", namespace,
 metric_name)
        except AttributeError:
            logger.warning("CloudWatch client is not available.")
```

Il obtient l'autorisation d'un rôle d'exécution que vous attribuez au cours du déploiement. Le rôle est défini dans le`aws-panorama-sample.yml` AWS CloudFormationmodèle.

Example aws-panorama-sample.yml

```
Resources:
  runtimeRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          -
            Effect: Allow
            Principal:
              Service:
                - panorama.amazonaws.com
            Action:
              - sts:AssumeRole
      Policies:
        - PolicyName: cloudwatch-putmetrics
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action: 'cloudwatch:PutMetricData'
                Resource: '*'
      Path: /service-role/
```

L'exemple d'application installe le SDK pour Python et d'autres dépendances avec pip. Lorsque vous créez le conteneur d'applications, le`Dockerfile`exécute des commandes pour installer des bibliothèques en plus de ce qui est fourni avec l'image de base.

Example Dockerfile

```
FROM public.ecr.aws/panorama/panorama-application
WORKDIR /panorama
COPY . .
RUN pip install --no-cache-dir --upgrade pip && \
    pip install --no-cache-dir -r requirements.txt
```

Pour utiliser le pluginAWSSDK dans le code de votre application, modifiez d'abord le modèle pour ajouter des autorisations pour toutes les actions d'API utilisées par l'application. Mettre à jour leAWS CloudFormationempiler en exécutant`1-create-role.sh`chaque fois que vous apportez une modification. Déployez ensuite les modifications apportées au code de votre application.

Pour les actions qui modifient ou utilisent des ressources existantes, il est recommandé de minimiser la portée de cette stratégie en spécifiant un nom ou un modèle pour la cible.`Resource`dans une déclaration séparée. Pour plus d'informations sur les actions et les ressources prises en charge par chaque service, consultezActions, ressources et clés de conditiondans le Service Authorization Reférence

## Étapes suivantes

Pour obtenir des instructions sur l'utilisation de l'interface de ligne de commande d'application AWS Panorama pour créer des applications et créer des packages à partir de zéro, consultez le README de l'interface

- github.com/aws/aws-panorama-cli

Pour obtenir d'autres exemples de code et un utilitaire de test que vous pouvez utiliser pour valider le code de votre application avant le déploiement, visitez le référentiel d'exemples AWS Panorama.

- github.com/aws-samples/aws-panorama-samples

# Modèles et caméras de vision par ordinateur pris en charge

AWS Panorama prend en charge les modèles PyTorch créés avec Apache MXNet etTensorFlow. Lorsque vous déployez une application, AWS Panorama compile votre modèle dans SageMaker Neo. Vous pouvez créer des modèles sur Amazon SageMaker ou dans votre environnement de développement, à condition d'utiliser des couches compatibles avec SageMaker Neo.

Pour traiter des vidéos et obtenir des images à envoyer à un modèle, l'appliance AWS Panorama se connecte à un flux vidéo codé H.264 à l'aide du protocole RTSP. AWS Panorama teste la compatibilité de diverses caméras courantes.

Sections



## Modèles pris en charge

Lorsque vous créez une application pour AWS Panorama, vous fournissez un modèle d'apprentissage automatique que l'application utilise pour la vision par ordinateur. Vous pouvez utiliser des modèles prédéfinis et préentraînés fournis par des cadres de modèles, un exemple de modèle ou un modèle que vous créez et entraînez vous-même.

> **Note**
>
> Pour obtenir la liste des modèles prédéfinis qui ont été testés avec AWS Panorama, consultez la section Compatibilité des modèles.

Lorsque vous déployez une application, AWS Panorama utilise le compilateur SageMaker Neo pour compiler votre modèle de vision par ordinateur. SageMakerNeo est un compilateur qui optimise les modèles pour qu'ils s'exécutent efficacement sur une plate-forme cible, qui peut être une instance d'Amazon Elastic Compute Cloud (Amazon EC2) ou un appareil périphérique tel que l'appliance AWS Panorama.

AWS Panorama prend en charge les versions d'PyTorchApache MXNet et celles TensorFlow qui sont prises en charge par SageMaker Neo pour les appareils de périphérie. Lorsque vous créez votre propre modèle, vous pouvez utiliser les versions du framework répertoriées dans les notes de mise à

SageMaker jour de Neo. DansSageMaker, vous pouvez utiliser l'algorithme de classification d'images intégré.

Pour plus d'informations sur l'utilisation de modèles dans AWS Panorama, consultezModèles de vision par ordinateur.

## Caméras compatibles

L'appliance AWS Panorama prend en charge les flux vidéo H.264 provenant de caméras émettant du RTSP sur un réseau local. Pour les flux de caméra supérieurs à 2 mégapixels, l'appliance réduit l'image à 1920 x 1080 pixels ou à une taille équivalente préservant le rapport hauteur/largeur du flux.

La compatibilité des modèles de caméra suivants avec l'appliance AWS Panorama a été testée :

- Axe : M3057-PLVE, M3058-PLVE, P1448-LE, P3225-LV Mk II
- LaView— LV-PB3040W
- Vivotek — IB9360-H
- Amcrest — IP2M-841B
- Application — IPC-B850W-S-3X, IPC-D250W-S
- WGCC — Dome PoE 4 mégapixels ONVIF

Pour les spécifications matérielles de l'appliance, reportez-vous à la sectionSpécifications de l'appliance AWS Panorama.

# Spécifications de l'appliance AWS Panorama

L'appliance AWS Panorama dispose des spécifications matérielles suivantes. Pour autresappareils compatibles, consultez la documentation du fabricant.

| Composant | Spécification |
| --- | --- |
| Processeur et GPU | Nvidia Jetson AGX Xavieravec 32 Go de RAM |
| Ethernet | 2 x 1000 Base-T (Gigabyte) |
| USB | 1 port USB 2.0 et 1 port USB 3.0 type A femelle |
| Sortie HDMI | 2,0 a |
| Dimensions | 7,75 po x 9,6 po x 1,6 po (197 mm x 243 mm x 40 mm) |
| Weight | 1,7 kg (3,7 lb) |
| Alimentation électrique | 100 V-240 V 50-60 Hz CA 65 W |
| Entrée d'alimentation | Prise IEC 60320 C6 (3 broches) |
| Protection contre la poussière et les liquides | IP-62 |
| Conformité réglementaire EMI/EMC | FAC Part-15 (États-Unis) |
| Limites thermiques tactiles | IEC-62368 |
| Température de fonctionnement | -20 °C à 60 °C |
| Humidité opérationnelle | 0 % à 95 % HR |
| Température de stockage | -20 °C à 85 °C |
| humidité de stockage | Non contrôlé pour basse température. 90 % HR à haute température |
| Refroidissement | Extraction thermique à air pulsé (ventilateur) |

| Composant | Spécification |
| --- | --- |
| Des options de montage | Montage en rack ou autoportant |
| Cordon d'alimentation | 1,8 mètre (6 pieds) |
| Contrôle de l'alimentation | Bouton-poussoir |
| Reset | Changement momentané |
| LED d'état et de réseau | LED RGB 3 couleurs programmable |

Le stockage Wi-Fi, Bluetooth et carte SD sont présents sur l'appliance, mais ils ne sont pas utilisables.

L'appliance AWS Panorama comprend deux vis pour le montage sur un rack de serveur. Vous pouvez monter deux appareils électroménagers side-by-side sur un rack de 19 pouces.

# Service Quotas

AWS Panorama applique des quotas aux ressources que vous créez dans votre compte et aux applications que vous déployez. Si vous utilisez AWS Panorama à plusieurs reprisesAWSRégions, les quotas s'appliquent séparément à chaque région. Les quotas AWS Panorama ne sont pas ajustables.

Les ressources d'AWS Panorama incluent les appareils, les packages de nœuds d'applications et les instances d'applications.

- Appareils— Jusqu'à 50 appareils enregistrés par région.
- Packages de nœuds— 50 packages par région, avec jusqu'à 20 versions par package.
- Instances d'applications— Jusqu'à 10 applications par appareil. Chaque application peut surveiller jusqu'à 8 flux de caméras. Les déploiements sont limités à 200 par jour pour chaque appareil.

Lorsque vous utilisez l'interface de ligne de commande de l'application AWS Panorama,AWS Command Line Interface, ouAWSSDK avec le service AWS Panorama, des quotas s'appliquent au nombre d'appels d'API que vous effectuez. Vous pouvez effectuer jusqu'à 5 demandes par seconde au total. Un sous-ensemble d'opérations d'API qui créent ou modifient des ressources applique une limite supplémentaire d'une demande par seconde.

Pour une liste complète des quotas, consultez leConsole Service Quotas, ou voirPoints de terminaison et quotas AWS Panoramadans leRéférence générale d'Amazon Web Services.

# Autorisations AWS Panorama

Vous pouvez utiliserAWS Identity and Access Management (IAM) pour gérer l'accès auAWS Panorama service et aux ressources telles les dispositifs et les applications. Pour les utilisateurs de votre compte qui utilisentAWS Panorama, vous gérez les autorisations dans une stratégie d'autorisations que vous pouvez appliquer aux rôles IAM. Pour gérer les autorisations d'une application, vous devez créer un rôle et l'attribuer à l'application.

Pour gérer les autorisations pour les utilisateurs dans votre compte, utilisez la stratégie gérée quiAWS Panorama fournit, ou écrivez votre propre stratégie. Vous devez disposer d'autorisations d'accès à d'autresAWS services pour obtenir les journaux des applications et des appliances, consulter les métriques et attribuer un rôle à une application.

UneAWS Panorama appliance est également dotée d'un rôle qui lui accorde l'autorisation d'accéder auxAWS services et aux ressources. Le rôle de l'appliance est l'un des rôles de service que leAWS Panorama service utilise pour accéder à d'autres services en votre nom.

Un rôle d'application est un rôle de service distinct que vous créez pour une application, afin de lui accorder l'autorisation d'utiliserAWS des services avec leAWS SDK for Python (Boto). Pour créer un rôle d'application, vous avez besoin de privilèges administratifs ou de l'aide d'un administrateur.

Vous pouvez restreindre les autorisations utilisateur en fonction de la ressource affectée par une action et, dans certains cas, en fonction de conditions supplémentaires. Par exemple, vous pouvez spécifier un modèle pour l'ARN (Amazon Resource Name) d'une application qui requiert qu'un utilisateur inclue son nom d'utilisateur dans le nom des applications qu'il crée. Pour connaître les ressources et les conditions prises en charge par chaque action, veuillez consulter Actions, Resources, and Condition Keys for Services (Référence des autorisations de service)AWS Panorama dans le document Service Authorization Reference (Référence des autorisations de service).

Pour plus d'informations, consultez Qu'est-ce qu'IAM ? dans le Guide de l'utilisateur IAM.

Rubriques

- Politiques IAM basées sur l'identité pour AWS Panorama
- Rôles de service et ressources interservices AWS Panorama
- Accorder des autorisations à une application

# Politiques IAM basées sur l'identité pour AWS Panorama

Pour accorder aux utilisateurs de votre compte l'accès à AWS Panorama, vous devez utiliser des stratégies basées sur une identité dansAWS Identity and Access Management (IAM). Appliquez des stratégies basées sur l'identité aux rôles IAM associés à un utilisateur. Vous pouvez également accorder aux utilisateurs d'un autre compte l'autorisation d'assumer un rôle dans votre compte et d'accéder à vos ressources AWS Panorama.

AWS Panorama fournit des stratégies gérées par qui accordent l'accès aux actions d'API d'AWS Panorama et, dans certains cas, l'accès à d'autres services utilisés pour développer et gérer des ressources AWS Panorama. AWS Panorama met à jour les stratégies gérées, les cas échéant, afin de garantir que vos utilisateurs ont accès aux nouvelles fonctionnalités lorsqu'elles sont disponibles.

- AWSPanoramaFullAccess— Fournit un accès complet à AWS Panorama, aux points d'accès AWS Panorama dans Amazon S3, aux informations d'identification et aux journaux de l'appliance sur Amazon CloudWatch.AWS Secrets Manager Inclut l'autorisation de créer un rôle lié à un service pour AWS Panorama. Afficher la politique

LaAWSPanoramaFullAccess politique vous permet de baliser les ressources AWS Panorama, mais ne dispose pas de toutes les autorisations liées aux balises utilisées par la console AWS Panorama. Pour accorder ces autorisations et, vous devez ajouter la politique suivante.

- ResourceGroupsandTagEditorFullAccess— Afficher la politique

LaAWSPanoramaFullAccess politique n'inclut pas l'autorisation d'acheter des appareils depuis la console AWS Panorama. Pour accorder ces autorisations et, vous devez ajouter la politique suivante.

- ElementalAppliancesSoftwareFullAccess— Afficher la politique

Les stratégies gérées par accordent une autorisation sur des actions d'API, sans restreindre les ressources qu'un utilisateur peut modifier. Pour bénéficier d'un contrôle plus précis, vous pouvez créer vos propres stratégies qui limitent la portée des autorisations d'un utilisateur. Utilisez la politique d'accès complet comme point de départ pour vos politiques.

> **Création de rôles de service**
>
> La première fois que vous utilisez la console AWS Panorama, vous devez être autorisé à créer le rôle de service utilisé par l'appliance AWS Panorama. Un rôle de service donne à un service l'autorisation de gérer des ressources ou d'interagir avec d'autres services. Créez ce rôle avant d'accorder l'accès à vos utilisateurs.

Pour plus d'informations sur les ressources et les conditions que vous pouvez utiliser pour limiter l'étendue des autorisations d'un utilisateur dans AWS Panorama, consultez Actions, ressources et clés de condition pour AWS Panorama dans la référence d'autorisation de service.

# Rôles de service et ressources interservices AWS Panorama

AWS Panorama utilise d'autres services AWS pour gérer l'appliance AWS Panorama, stocker des données et importer des ressources d'application. Un rôle de service donne à un service l'autorisation de gérer des ressources ou d'interagir avec d'autres services. Lorsque vous vous connectez à la console AWS Panorama pour la première fois, vous créez les rôles de service suivants :

- AWSServiceRoleForAWSPanorama— Permet à AWS Panorama de gérer les ressources dans AWS IoT, AWS Secrets Manager et AWS Panorama.

  Stratégie gérée :AWSPanoramaServiceLinkedRolePolicy
- AWSPanoramaApplianceServiceRole— Permet à une appliance AWS Panorama de télécharger des journaux vers CloudWatch, et pour obtenir des objets depuis les points d'accès Amazon S3 créés par AWS Panorama.

  Stratégie gérée :AWSPanoramaApplianceServiceRolePolicy

Pour consulter les autorisations associées à chaque rôle, utilisez leConsole IAM. Dans la mesure du possible, les autorisations du rôle sont limitées aux ressources qui correspondent à un modèle de dénomination utilisé par AWS Panorama. Par exemple,`AWSServiceRoleForAWSPanorama`accorde uniquement l'autorisation d'accès au serviceAWS IoTdes ressources qui ont`panorama`en leur nom.

Sections



## Sécurisation du rôle d'appliance

L'appliance AWS Panorama utilise le`AWSPanoramaApplianceServiceRole`rôle permettant d'accéder aux ressources de votre compte. L'appliance est autorisée à charger des journaux sur CloudWatch Journaux, lecture des informations d'identification du flux deAWS Secrets Manager, et pour accéder aux artefacts d'application dans les points d'accès Amazon Simple Storage Service (Amazon S3) créés par AWS Panorama.

> **Note**
>
> Les applications n'utilisent pas les autorisations de l'appliance. Pour autoriser votre application à utiliserAWSservices, créez unrôle dans l'application.

AWS Panorama utilise le même rôle de service avec toutes les appliances de votre compte et n'utilise pas de rôles entre les comptes. Pour renforcer la sécurité, vous pouvez modifier la politique de confiance du rôle d'appliance afin de l'appliquer de manière explicite, ce qui constitue une bonne pratique lorsque vous utilisez des rôles pour accorder à un service l'autorisation d'accéder aux ressources de votre compte.

Pour mettre à jour la politique de confiance relative aux rôles

1. Ouvrez le rôle d'appliance dans la console IAM :AWSPanoramaApplianceServiceRole
2. Choisissez Modifier la relation d'approbation.
3. Mettez à jour le contenu de la politique, puis choisissezMettre la politique d'approbation.

La politique de confiance suivante inclut une condition qui garantit que lorsqu'AWS Panorama assume le rôle d'appliance, il le fait pour une appliance de votre compte. Dans la`aws:SourceAccount`Une condition compare l'ID de compte spécifié par AWS Panorama à celui que vous incluez dans la politique.

Example politique de confiance — Compte spécifique

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "panorama.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
```

```
  ]
}
```

Si vous souhaitez restreindre davantage AWS Panorama et lui permettre de jouer le rôle uniquement sur un appareil spécifique, vous pouvez spécifier l'appareil par ARN. Dans laaws:SourceArnUne condition compare l'ARN de l'appliance spécifiée par AWS Panorama à celui que vous incluez dans la politique.

Example politique de confiance — Appliance unique

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "panorama.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:panorama:us-east-1:123456789012:device/
device-lk7exmplpvcr3heqwjmesw76ky"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Si vous réinitialisez et reprovisionnez l'appliance, vous devez supprimer temporairement la condition ARN source, puis l'ajouter à nouveau avec le nouvel identifiant de l'appareil.

Pour en savoir plus sur ces conditions et sur les bonnes pratiques en matière de sécurité lorsque des services utilisent des rôles pour accéder aux ressources de votre compte, consultezLe problème du député confusdans le Guide de l'utilisateur IAM.

## Utilisation d'autres services

AWS Panorama crée ou accède à des ressources dans les services suivants :

- AWS IoT— Objets, politiques, certificats et tâches pour l'appliance AWS Panorama
- Amazon S3— Points d'accès pour la mise en place de modèles d'applications, de code et de configurations.
- Secrets Manager— Informations d'identification à court terme pour l'appliance AWS Panorama.

Pour plus d'informations sur le format Amazon Resource Name (ARN) ou les limites d'autorisation pour chaque service, consultez les rubriques duIAM User Guidequi sont liés dans cette liste.

# Accorder des autorisations à une application

Vous pouvez créer un rôle pour votre application afin de lui accorder l'autorisation d'appelerAWSServices . Par défaut, les applications ne disposent pas d'autorisations. Vous créez un rôle d'application dans IAM et vous l'affectez à une application pendant le déploiement. Pour accorder à votre application uniquement les autorisations dont elle a besoin, créez un rôle avec des autorisations pour des actions API spécifiques.

LeExemple d'applicationinclut unAWS CloudFormationmodèle et script qui créent un rôle d'application. C'est unRôle de servicequ'peut endosser AWS Panorama. Ce rôle accorde l'autorisation à l'application d'appeler CloudWatch pour charger des mesures.

Example aws-panorama-sample.yml— Rôle d'application

```
Resources:
  runtimeRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          -
            Effect: Allow
            Principal:
              Service:
                - panorama.amazonaws.com
            Action:
              - sts:AssumeRole
      Policies:
        - PolicyName: cloudwatch-putmetrics
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action: 'cloudwatch:PutMetricData'
                Resource: '*'
      Path: /service-role/
```

Vous pouvez étendre ce script pour accorder des autorisations à d'autres services, en spécifiant une liste d'actions ou de modèles d'API pour la valeur de`Action`.

Pour plus d'informations sur les autorisations dans AWS Panorama, consultezAutorisations AWS Panorama.

# Gestion duAWS PanoramaAppliance

LeAWS PanoramaL'appliance est le matériel qui exécute vos applications. Vous utilisez leAWS Panoramapour enregistrer une appliance, mettre à jour son logiciel et y déployer des applications. Le logicielAWS PanoramaL'appliance se connecte aux flux de caméras, envoie des images vidéo à votre application et affiche la sortie vidéo sur un écran connecté.

Après avoir configuré votre appliance ou un autreappareil compatible, vous enregistrez des caméras pour une utilisation avec des applications. VousGérer les flux de camérasdans leAWS Panoramaconsole Lorsque vous déployez une application, vous choisissez les flux de caméra que l'appliance lui envoie pour traitement.

Pour les didacticiels qui présentent leAWS PanoramaAppliance avec exemple d'application, voirDémarrer avec AWS Panorama.

Rubriques

- Gestion d'une appliance AWS Panorama
- Connexion de l'appliance AWS Panorama à votre réseau
- Gestion des flux de caméras dans AWS Panorama
- Gérer les applications sur une appliance AWS Panorama
- Boutons et voyants de l'appliance AWS Panorama

# Gestion d'une appliance AWS Panorama

Vous utilisez la console AWS Panorama pour configurer, mettre à niveau ou annuler l'enregistrement de l'appliance AWS Panorama et d'autres appareils compatibles.

Pour configurer une appliance, suivez les instructions du didacticiel de mise en route. Le processus de configuration crée les ressources dans AWS Panorama qui permettent de suivre votre appliance et de coordonner les mises à jour et les déploiements.

Pour enregistrer une appliance auprès de l'API AWS Panorama, consultezAutomatisez l'enregistrement des appareils.

Sections

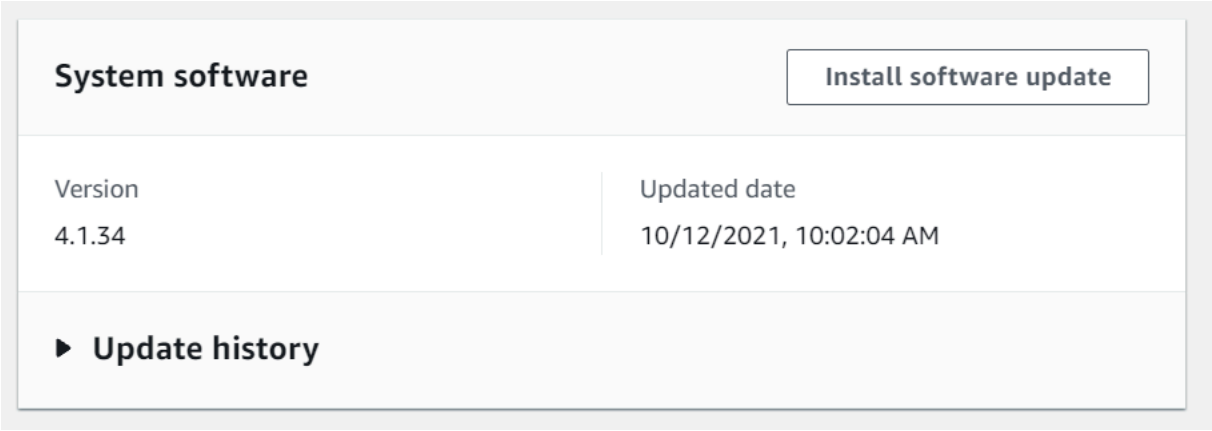


## Mettre à jour le logiciel de l'appliance

Vous visualisez et déployez les mises à jour logicielles pour l'appliance dans la console AWS Panorama. Les mises à jour peuvent être obligatoires ou facultatives. Lorsqu'une mise à jour requise est disponible, la console vous invite à l'appliquer. Vous pouvez appliquer des mises à jour facultatives sur la page des paramètres de l'appliance.

Pour mettre à jour le logiciel de l'appliance

1. Ouvrez la page Appareils de la console AWS Panorama.
2. Choisissez un appareil.
3. Choisissez Réglages
4. Sous Logiciel système, choisissez Installer la mise à jour logicielle.




5. Choisissez une nouvelle version, puis choisissez Installer.

## Annuler l'enregistrement d'un appareil

Si vous avez fini de travailler avec une appliance, vous pouvez utiliser la console AWS Panorama pour la désenregistrer et supprimer lesAWS IoT ressources associées.

Pour supprimer un appareil

1. Ouvrez la page Appareils de la console AWS Panorama.
2. Choisissez le nom de l'appliance.
3. Sélectionnez Delete (Supprimer).
4. Entrez le nom de l'appliance et choisissez Supprimer.

Lorsque vous supprimez une appliance du service AWS Panorama, les données de l'appliance ne sont pas supprimées automatiquement. Un appareil désenregistré ne peut pas se connecter auxAWS services et ne peut pas être réenregistré tant qu'il n'est pas réinitialisé.

## Redémarrer une appliance

Vous pouvez redémarrer une appliance à distance.

Pour redémarrer une appliance

1. Ouvrez la page Appareils de la console AWS Panorama.
2. Choisissez le nom de l'appliance.
3. Choisissez Redémarrer.

La console envoie un message à l'appliance pour qu'elle redémarre. Pour pouvoir recevoir le signal, l'appareil doit pouvoir se connecter àAWS IoT. Pour redémarrer une appliance à l'aide de l'API AWS Panorama, consultezRedémarrer les dispositifs.

## Réinitialiser une appliance

Pour utiliser une appliance dans une autre région ou avec un compte différent, vous devez la réinitialiser et la réapprovisionner avec un nouveau certificat. La réinitialisation de l'appareil applique la dernière version logicielle requise et supprime toutes les données du compte.

Pour démarrer une opération de réinitialisation, l'appareil doit être branché et mis hors tension. Appuyez sur les boutons d'alimentation et de réinitialisation et maintenez-les enfoncés pendant cinq secondes. Lorsque vous relâchez les boutons, le voyant d'état clignote en orange. Attendez que le voyant d'état clignote en vert avant de mettre en service ou de déconnecter l'appliance.

Vous pouvez également réinitialiser le logiciel de l'appliance sans supprimer les certificats de l'appareil. Pour plus d'informations, consultez Boutons d'alimentation et de réinitialisation.

# Connexion de l'appliance AWS Panorama à votre réseau

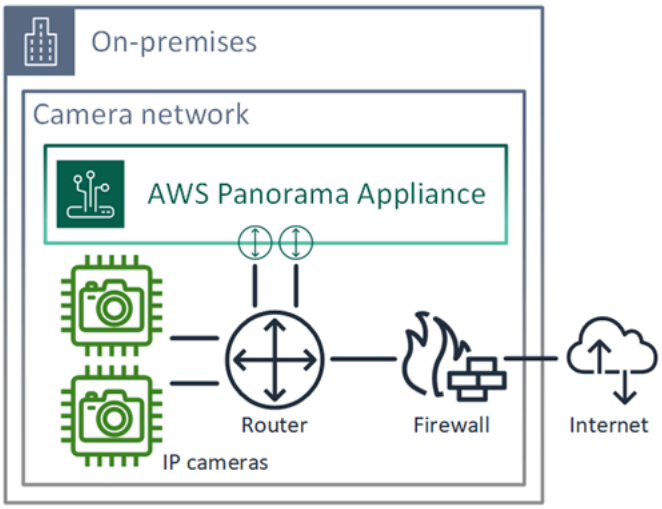
L'appliance AWS Panorama nécessite une connectivité au AWS cloud et à votre réseau de caméras IP sur site. Vous pouvez connecter l'appliance à un pare-feu unique qui autorise l'accès aux deux, ou connecter chacune des deux interfaces réseau de l'appareil à un sous-réseau différent. Dans les deux cas, vous devez sécuriser les connexions réseau de l'appliance pour empêcher tout accès non autorisé aux flux de vos caméras.

Sections



## Configuration réseau unique

L'appliance possède deux ports Ethernet. Si vous acheminez tout le trafic vers et depuis le périphérique via un seul routeur, vous pouvez utiliser le deuxième port pour assurer la redondance au cas où la connexion physique au premier port serait interrompue. Configurez votre routeur pour autoriser l'appliance à se connecter uniquement aux flux de caméras et à Internet, et pour empêcher les flux de caméras de quitter votre réseau interne.

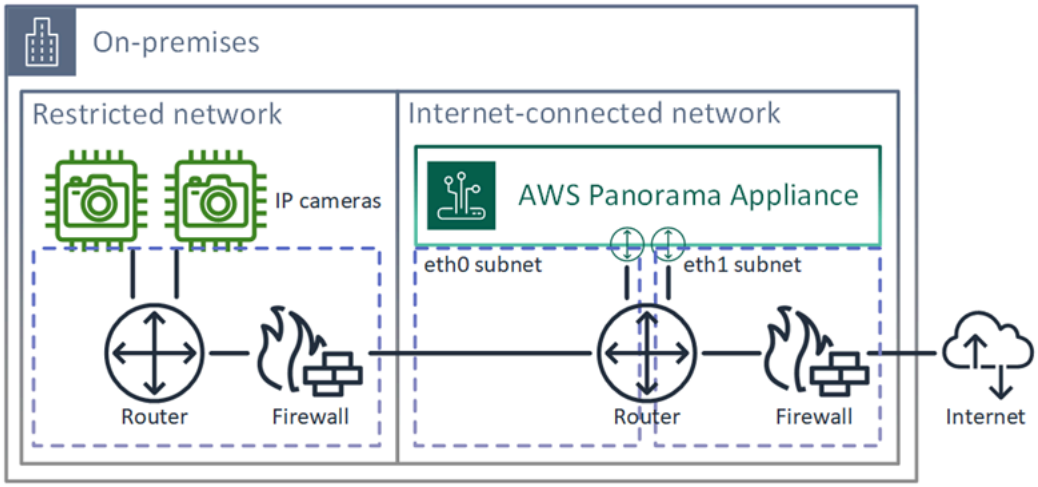


Pour plus de détails sur les ports et les points de terminaison auxquels l'appliance doit accéder, reportez-vous aux sections Configuration de l'accès aux services etConfiguration de l'accès au réseau local.

## Configuration réseau double

Pour un niveau de sécurité supplémentaire, vous pouvez placer l'appliance sur un réseau connecté à Internet distinct de votre réseau de caméras. Un pare-feu entre votre réseau de caméras restreint et le réseau de l'appliance permet uniquement à l'appliance d'accéder aux flux vidéo. Si votre réseau de caméras était auparavant isolé pour des raisons de sécurité, vous préférerez peut-être cette méthode plutôt que de connecter le réseau de caméras à un routeur qui autorise également l'accès à Internet.

L'exemple suivant montre que l'appliance se connecte à un sous-réseau différent sur chaque port. Le routeur place l'`eth0`interface sur un sous-réseau qui route vers le réseau de caméras, et `eth1` sur un sous-réseau qui route vers Internet.




Vous pouvez confirmer l'adresse IP et l'adresse MAC de chaque port dans la console AWS Panorama.

## Configuration de l'accès aux services

Pendant le provisionnement, vous pouvez configurer l'appliance pour demander une adresse IP spécifique. Choisissez une adresse IP à l'avance pour simplifier la configuration du pare-feu et garantir que l'adresse de l'appliance ne change pas si elle est hors ligne pendant une longue période.

L'appliance utilise des AWS services pour coordonner les mises à jour logicielles et les déploiements. Configurez votre pare-feu pour permettre à l'appliance de se connecter à ces points de terminaison.

Accès Internet

- AWS IoT(HTTPS et MQTT, ports 443, 8443 et 8883) — et les points de terminaison de gestion AWS IoT Core des appareils. Pour plus de détails, consultez la section Points de terminaison et quotas AWS IoT Device Management dans leRéférence générale d'Amazon Web Services.
- AWS IoTinformations d'identification (HTTPS, port 443) — `credentials.iot.<region>.amazonaws.com` et sous-domaines.
- Amazon Elastic Container Registry (HTTPS, port 443) — `api.ecr.<region>.amazonaws.com` `dkr.ecr.<region>.amazonaws.com` et sous-domaines.
- Amazon CloudWatch (HTTPS, port 443) —`monitoring.<region>.amazonaws.com`.
- Amazon CloudWatch Logs (HTTPS, port 443) —`logs.<region>.amazonaws.com`.
- Amazon Simple Storage Service (HTTPS, port 443) — `s3.<region>.amazonaws.com` `s3-accesspoint.<region>.amazonaws.com` et sous-domaines.

Si votre application appelle d'autres AWS services, l'appliance doit également accéder aux points de terminaison de ces services. Pour plus d'informations, consultez la section Points de terminaison et quotas du service.

## Configuration de l'accès au réseau local

L'appliance doit accéder aux flux vidéo RTSP localement, mais pas via Internet. Configurez votre pare-feu pour autoriser l'appliance à accéder aux flux RTSP sur le port 554 en interne, et pour empêcher les flux de sortir ou d'entrer depuis Internet.

Accès local

- Protocole de diffusion en temps réel (RTSP, port 554) — Pour lire les flux de caméras.
- Protocole horaire réseau (NTP, port 123) : pour synchroniser l'horloge de l'appliance. Si vous n'utilisez pas de serveur NTP sur votre réseau, l'appliance peut également se connecter à des serveurs NTP publics via Internet.

## Connectivité privée

L'appliance AWS Panorama n'a pas besoin d'accès à Internet si vous la déployez dans un sous-réseau VPC privé avec une connexion VPN à. AWS Vous pouvez utiliser le VPN Site-to-Site ou AWS Direct Connect créer une connexion VPN entre un routeur local et. AWS Au sein de votre sous-

réseau VPC privé, vous créez des points de terminaison qui permettent à l'appliance de se connecter à Amazon Simple Storage Service et à d'autres AWS IoT services. Pour plus d'informations, consultez Connexion d'une appliance à un sous-réseau privé.

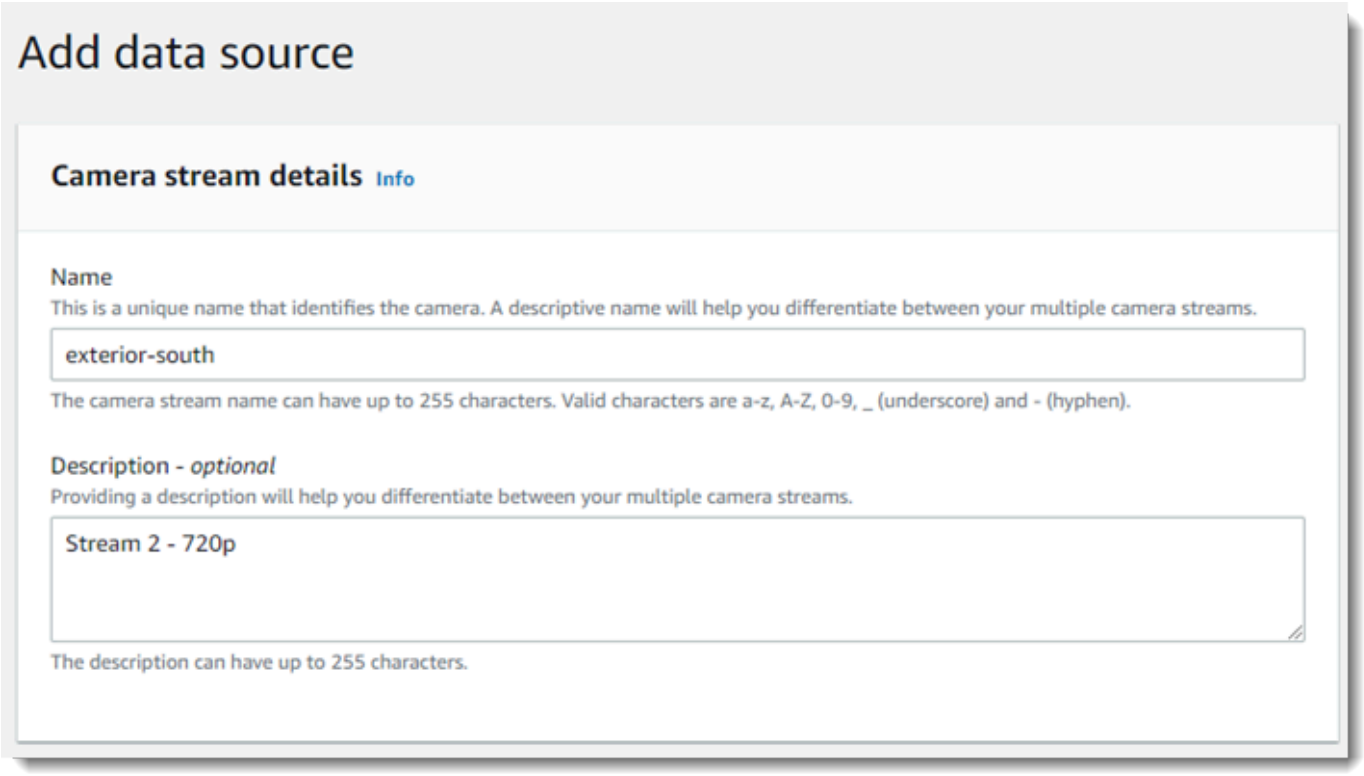
# Gestion des flux de caméras dans AWS Panorama

Pour enregistrer des flux vidéo en tant que sources de données pour votre application, utilisez la console AWS Panorama. Une application peut traiter plusieurs flux simultanément et plusieurs appliances peuvent se connecter au même flux.

> **Important**
>
> Une application peut se connecter à n'importe quel flux de caméra routable à partir du réseau local auquel elle se connecte. Pour sécuriser vos flux vidéo, configurez votre réseau pour autoriser uniquement le trafic RTSP localement. Pour plus d'informations, consultez Services de sécurité AWS Panorama.

Pour enregistrer un flux de caméra

1. Ouvrez la console AWS PanoramaPage Sources de données.
2. ChoisissezAjouter une source de données.




3. 以下を設定します。

- Nom— Un nom pour le flux de caméra.
- Description— Une brève description de l'appareil photo, de son emplacement ou d'autres détails.
- URL RTSP: URL qui spécifie l'adresse IP de la caméra et le chemin d'accès au flux. Par exemple, `rtsp://192.168.0.77/live/mpeg4/`
- Informations d'identification— Si le flux de caméra est protégé par mot de passe, spécifiez le nom d'utilisateur et le mot de passe.

4. Choisissez Save (Enregistrer).

Pour enregistrer un flux de caméras auprès de l'API AWS Panorama, reportez-vous à la sectionAutomatisez l'enregistrement des appareils.

Pour obtenir la liste des caméras compatibles avec AWS Panorama Appliance, voirModèles et caméras de vision par ordinateur pris en charge.

## Suppression d'un flux

Vous pouvez supprimer un flux de caméra dans la console AWS Panorama.

Pour supprimer un flux de caméra

1. Ouvrez la console AWS PanoramaPage Sources de données.
2. Choisissez un flux de caméra.
3. ChoisissezSupprimer une source de données.

La suppression d'un flux de caméra du service n'arrête pas l'exécution d'applications ni ne supprime les informations d'identification de la caméra de Secrets Manager. Pour supprimer des secrets, utilisez leConsole Secrets Manager.

# Gérer les applications sur une appliance AWS Panorama

Une application est une combinaison de code, de modèles et de configuration. De laAppareilsdans la console AWS Panorama, vous pouvez gérer des applications sur l'appliance.

Pour gérer des applications sur une appliance AWS Panorama

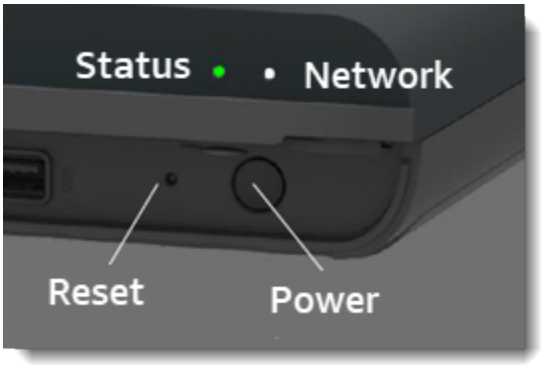
1. Ouvrez la console AWS PanoramaPage des appareils.
2. Choisissez une appliance.

LeApplications déployaffiche les applications qui ont été déployées sur l'appliance.

Utilisez les options de cette page pour supprimer des applications déployées de l'appliance ou remplacer une application en cours d'exécution par une nouvelle version. Vous pouvez également cloner une application (en cours d'exécution ou supprimée) pour en déployer une nouvelle copie.

# Boutons et voyants de l'appliance AWS Panorama

L'appliance AWS Panorama possède deux voyants LED au-dessus du bouton d'alimentation qui indiquent l'état de l'appareil et la connectivité réseau.




## Lampe d'état

Les LED changent de couleur et clignotent pour indiquer l'état. Un clignotement lent se produit une fois toutes les trois secondes. Un clignotement rapide se fait une fois par seconde.

État de la LED

- Fast Clign vert— L'appliance est en train de démarrer.
- Vert uni— L'appareil fonctionne normalement.
- Bleu qui clignote lents— L'appliance copie les fichiers de configuration et tente de s'enregistrer auprèsAWS IoT.
- Fast Clign Bleu/Fast— L'appareil estcopie d'une image de journalsur une clé USB.
- Fast Clign rouge— L'appareil a rencontré une erreur lors du démarrage ou est en surchauffe.
- Vert clignotement lent— L'appliance est en train de restaurer la dernière version du logiciel.
- Fast Clign— L'appliance restaure la version minimale du logiciel.

## Éclairage réseau

Les états de la LED réseau sont les suivants :

États des LED du réseau

- Vert uni— Un câble Ethernet est connecté.

- Vert clignotant— L'appliance communique via le réseau.
- Rouge uni— Aucun câble Ethernet n'est connecté.

## Boutons d'alimentation et de réinitialisation

Les boutons d'alimentation et de réinitialisation se trouvent à l'avant de l'appareil, sous un couvercle de protection. Le bouton de réinitialisation est plus petit et encastré. Utilisez un petit tournevis ou un trombone pour appuyer dessus.

Pour réinitialiser un appareil

1. L'appareil doit être branché et éteint. Pour mettre l'appareil hors tension, maintenez le bouton d'alimentation enfoncé pendant 1 seconde et attendez la fin de la séquence d'arrêt. La séquence d'arrêt dure environ 10 secondes.
2. Pour réinitialiser l'appareil, utilisez les combinaisons de boutons suivantes. Un appui court dure 1 seconde. Un appui long dure 5 secondes. Pour les opérations nécessitant plusieurs boutons, appuyez simultanément sur les deux boutons et maintenez-les enfoncés.

   - Fast Insert— Appuyez longuement sur l'alimentation et réinitialisez.

     Restaure la version minimale du logiciel et supprime tous les fichiers de configuration et toutes les applications.
   - Restaurer la dernière version du logiciel— Appuyez brièvement sur Reset.

     Réapplique la dernière mise à jour logicielle à l'appliance.
   - Restaurer la version minimale du logiciel— Appuyez longuement sur Reset.

     Réapplique la dernière mise à jour logicielle requise à l'appliance.
3. Relâchez les deux boutons. L'appareil s'allume et le voyant d'état clignote en orange pendant plusieurs minutes.
4. Lorsque l'appliance est prête, le voyant d'état clignote en vert.

La réinitialisation d'une appliance ne la supprime pas du service AWS Panorama. Pour plus d'informations, consultez Annuler l'enregistrement d'un appareil.

# Gestion d'AWS Panoramaapplications

Applications exécutées surAWS PanoramaAppareil permettant d'effectuer des tâches de vision par ordinateur sur des flux vidéo. Vous pouvez créer des applications de vision par ordinateur en combinant du code Python et des modèles d'apprentissage automatique, puis les déployer surAWS PanoramaAppareil sur Internet. Les applications peuvent envoyer des vidéos sur un écran ou utiliser le kit SDK AWS pour envoyer les résultats aux services AWS.

Rubriques

# Déployer une application

Pour déployer une application, vous utilisez l'interface de ligne de commande de l'application AWS Panorama, vous l'importez dans votre compte, vous créez le conteneur, vous chargez et enregistrez des actifs et vous créez une instance d'application. Cette rubrique décrit chacune de ces étapes en détail et décrit ce qui se passe en arrière-plan.

Si vous n'avez pas encore déployé d'application, consultez la procédure pas Démarrer avec AWS Panorama à pas.

Pour plus d'informations sur la personnalisation et l'extension de l'exemple d'application, reportez-vous Création d'applications AWS Panorama à la section.

Sections

- Installation de l'interface de ligne de commande de l'application AWS Panorama
- Importer une application
- Création d'une image de conteneur
- Importer un modèle
- Charger les actifs de l'application
- Déployer une application avec la console AWS Panorama
- Automatisez le déploiement des applications

## Installation de l'interface de ligne de commande de l'application AWS Panorama

Pour installer l'interface de ligne de commande de l'application AWS PanoramaAWS CLI, utilisez pip.

```
$ pip3 install --upgrade awscli panoramacli
```

Pour créer des images d'application à l'aide de l'interface de ligne de commande AWS Panorama Application, vous avez besoin de Docker. Sous Linux, `qemu` les bibliothèques système associées sont également requises. Pour plus d'informations sur l'installation et la configuration de l'interface de ligne de commande de l'application AWS Panorama, consultez le fichier README dans le référentiel du GitHub projet.

- github.com/aws/ aws-panorama-cli

Pour obtenir des instructions sur la configuration d'un environnement de génération dans Windows avec WSL2, consultez. Configuration d'un environnement de développement sous Windows

## Importer une application

Si vous travaillez avec un exemple d'application ou une application fournie par un tiers, utilisez l'interface de ligne de commande AWS Panorama Application pour importer l'application.

```
my-app$ panorama-cli import-application
```

Cette commande renomme les packages d'applications à l'aide de votre identifiant de compte. Les noms des packages commencent par l'ID du compte sur lequel ils sont déployés. Lorsque vous déployez une application sur plusieurs comptes, vous devez importer et empaqueter l'application séparément pour chaque compte.

Par exemple, l'exemple d'application de ce guide, un package de code et un package modèle, chacun étant nommé par un identifiant de compte réservé. La `import-application` commande les renomme pour utiliser l'ID de compte que la CLI déduit des informations d'identification de votre espace de travail. AWS

```
/aws-panorama-sample
### assets
### graphs
#   ### my-app
#       ### graph.json
### packages
    ### 123456789012-SAMPLE_CODE-1.0
    #   ### Dockerfile
    #   ### application.py
    #   ### descriptor.json
    #   ### package.json
    #   ### requirements.txt
    #   ### squeezenet_classes.json
    ### 123456789012-SQUEEZENET_PYTORCH-1.0
        ### descriptor.json
        ### package.json
```

`123456789012`est remplacé par votre identifiant de compte dans les noms des répertoires des packages et dans le manifeste de l'application (`graph.json`), qui y fait référence. Vous pouvez confirmer l'identifiant de votre compte `aws sts get-caller-identity` en appelant leAWS CLI.

```
$ aws sts get-caller-identity
{
    "UserId": "AIDAXMPL7W66UC3GFXMPL",
    "Account": "210987654321",
    "Arn": "arn:aws:iam::210987654321:user/devenv"
}
```

## Création d'une image de conteneur

Le code de votre application est empaqueté dans une image de conteneur Docker, qui inclut le code de l'application et les bibliothèques que vous installez dans votre Dockerfile. Utilisez la `build-container` commande CLI de l'application AWS Panorama pour créer une image Docker et exporter une image de système de fichiers.

```
my-app$ panorama-cli build-container --container-asset-name code_asset --package-path
 packages/210987654321-SAMPLE_CODE-1.0
{
    "name": "code_asset",
    "implementations": [
        {
            "type": "container",
            "assetUri":
 "5fa5xmplbc8c16bf8182a5cb97d626767868d3f4d9958a4e49830e1551d227c5.tar.gz",
            "descriptorUri":
 "1872xmpl129481ed053c52e66d6af8b030f9eb69b1168a29012f01c7034d7a8f.json"
        }
    ]
}
Container asset for the package has been succesfully built at
 assets/5fa5xmplbc8c16bf8182a5cb97d626767868d3f4d9958a4e49830e1551d227c5.tar.gz
```

Cette commande crée une image Docker nommée `code_asset` et exporte un système de fichiers vers une `.tar.gz` archive du `assets` dossier. La CLI extrait l'image de base de l'application depuis Amazon Elastic Container Registry (Amazon ECR), comme spécifié dans le Dockerfile de l'application.

Outre l'archive du conteneur, la CLI crée un actif pour le descripteur de package ()`descriptor.json`. Les deux fichiers sont renommés avec un identifiant unique qui reflète un hachage du fichier d'origine. L'interface de ligne de commande de l'application AWS Panorama

ajoute également un bloc à la configuration du package qui enregistre les noms des deux actifs. Ces noms sont utilisés par l'appliance au cours du processus de déploiement.

Example Packages/123456789012-sample_code-1.0/package.json — avec bloc de ressources

```
{
    "nodePackage": {
        "envelopeVersion": "2021-01-01",
        "name": "SAMPLE_CODE",
        "version": "1.0",
        "description": "Computer vision application code.",
        "assets": [
            {
                "name": "code_asset",
                "implementations": [
                    {
                        "type": "container",
                        "assetUri":
 "5fa5xmplbc8c16bf8182a5cb97d626767868d3f4d9958a4e49830e1551d227c5.tar.gz",
                        "descriptorUri":
 "1872xmpl129481ed053c52e66d6af8b030f9eb69b1168a29012f01c7034d7a8f.json"
                    }
                ]
            }
        ],
        "interfaces": [
            {
                "name": "interface",
                "category": "business_logic",
                "asset": "code_asset",
                "inputs": [
                    {
                        "name": "video_in",
                        "type": "media"
                    },
```

Le nom de la ressource de code, spécifié dans la `build-container` commande, doit correspondre à la valeur du `asset` champ dans la configuration du package. Dans l'exemple précédent, les deux valeurs sont`code_asset`.

## Importer un modèle

Il se peut que votre application contienne une archive de modèles dans son dossier de ressources ou que vous téléchargiez séparément. Si vous avez un nouveau modèle, un modèle mis à jour ou un fichier descripteur de modèle mis à jour, utilisez la `add-raw-model` commande pour l'importer.

```
my-app$ panorama-cli add-raw-model --model-asset-name model_asset \
      --model-local-path my-model.tar.gz \
      --descriptor-path packages/210987654321-SQUEEZENET_PYTORCH-1.0/descriptor.json \
      --packages-path packages/210987654321-SQUEEZENET_PYTORCH-1.0
```

Si vous avez juste besoin de mettre à jour le fichier descripteur, vous pouvez réutiliser le modèle existant dans le répertoire des actifs. Vous devrez peut-être mettre à jour le fichier descripteur pour configurer des fonctionnalités telles que le mode de précision à virgule flottante. Par exemple, le script suivant montre comment procéder avec l'exemple d'application.

Example scripts/utiles/ .sh update-model-config

```
#!/bin/bash
set -eo pipefail
MODEL_ASSET=fd1axmplacc3350a5c2673adacffab06af54c3f14da6fe4a8be24cac687a386e
MODEL_PACKAGE=SQUEEZENET_PYTORCH
ACCOUNT_ID=$(ls packages | grep -Eo '[0-9]{12}' | head -1)
panorama-cli add-raw-model --model-asset-name model_asset --model-local-path assets/
${MODEL_ASSET}.tar.gz --descriptor-path packages/${ACCOUNT_ID}-${MODEL_PACKAGE}-1.0/
descriptor.json --packages-path packages/${ACCOUNT_ID}-${MODEL_PACKAGE}-1.0
cp packages/${ACCOUNT_ID}-${MODEL_PACKAGE}-1.0/package.json packages/${ACCOUNT_ID}-
${MODEL_PACKAGE}-1.0/package.json.bup
```

Les modifications apportées au fichier descripteur dans le répertoire du package de modèles ne sont appliquées que lorsque vous le réimportez avec l'interface de ligne de commande. La CLI met à jour la configuration du package modèle avec les nouveaux noms de ressources en place, de la même manière qu'elle met à jour la configuration du package de code d'application lorsque vous reconstruisez un conteneur.

## Charger les actifs de l'application

Pour charger et enregistrer les actifs de l'application, notamment l'archive du modèle, l'archive du système de fichiers du conteneur et leurs fichiers descripteurs, utilisez la `package-application` commande.

```
my-app$ panorama-cli package-application
Uploading package SQUEEZENET_PYTORCH
Patch version for the package
 5d3cxmplb7113faa1d130f97f619655d8ca12787c751851a0e155e50eb5e3e96
Deregistering previous patch version
 e845xmpl8ea0361eb345c313a8dded30294b3a46b486dc8e7c174ee7aab29362
Asset fd1axmplacc3350a5c2673adacffab06af54c3f14da6fe4a8be24cac687a386e.tar.gz already
 exists, ignoring upload
upload: assets/87fbxmpl6f18aeae4d1e3ff8bbc6147390feaf47d85b5da34f8374974ecc4aaf.json
 to s3://arn:aws:s3:us-east-2:212345678901:accesspoint/
panorama-210987654321-6k75xmpl2jypelgzst7uux62ye/210987654321/nodePackages/
SQUEEZENET_PYTORCH/
binaries/87fbxmpl6f18aeae4d1e3ff8bbc6147390feaf47d85b5da34f8374974ecc4aaf.json
Called register package version for SQUEEZENET_PYTORCH with patch version
 5d3cxmplb7113faa1d130f97f619655d8ca12787c751851a0e155e50eb5e3e96
...
```

Si aucune modification n'est apportée à un fichier de ressources ou à la configuration du package, la CLI l'ignore.

```
Uploading package SAMPLE_CODE
Patch Version ca91xmplca526fe3f07821fb0c514f70ed0c444f34cb9bd3a20e153730b35d70 already
 registered, ignoring upload
Register patch version complete for SQUEEZENET_PYTORCH with patch version
 5d3cxmplb7113faa1d130f97f619655d8ca12787c751851a0e155e50eb5e3e96
Register patch version complete for SAMPLE_CODE with patch version
 ca91xmplca526fe3f07821fb0c514f70ed0c444f34cb9bd3a20e153730b35d70
All packages uploaded and registered successfully
```

La CLI télécharge les ressources de chaque package vers un point d'accès Amazon S3 spécifique à votre compte. AWS Panorama gère le point d'accès pour vous et fournit des informations à ce sujet via l'DescribePackageAPI. La CLI télécharge les ressources de chaque package vers l'emplacement prévu pour ce package et les enregistre auprès du service AWS Panorama avec les paramètres décrits dans la configuration du package.

## Déployer une application avec la console AWS Panorama

Vous pouvez déployer une application à l'aide de la console AWS Panorama. Au cours du processus de déploiement, vous choisissez les flux de caméra à transmettre au code de l'application et vous configurez les options fournies par le développeur de l'application.

Pour déployer une application

1. Ouvrez la page Applications déployées de la console AWS Panorama.
2. Choisissez Déployer l'application.
3. Collez le contenu du manifeste de l'application`graph.json`, dans l'éditeur de texte. Choisissez Suivant.
4. Entrez un nom et une description.
5. Choisissez Proceed to deploy.
6. Choisissez Commencer le déploiement.
7. Si votre application utilise un rôle, sélectionnez-le dans le menu déroulant. Choisissez Suivant.
8. Choisissez Sélectionner l'appareil, puis choisissez votre appareil. Choisissez Suivant.
9. À l'étape Sélectionner les sources de données, choisissez Afficher les entrées et ajoutez le flux de votre caméra en tant que source de données. Choisissez Suivant.
10. À l'étape Configurer, configurez tous les paramètres spécifiques à l'application définis par le développeur. Choisissez Suivant.
11. Choisissez Déployer, puis cliquez sur Terminé.
12. Dans la liste des applications déployées, choisissez l'application dont vous souhaitez contrôler l'état.

Le processus de déploiement prend 15 à 20 minutes. La sortie de l'appliance peut rester vide pendant une période prolongée pendant le démarrage de l'application. Si vous rencontrez une erreur, reportez-vous à la sectionRésolution des problèmes.

## Automatisez le déploiement des applications

Vous pouvez automatiser le processus de déploiement des applications à l'aide de l'CreateApplicationInstanceAPI. L'API prend deux fichiers de configuration en entrée. Le manifeste de l'application précise les packages utilisés et leurs relations. Le second fichier est un fichier de remplacement qui spécifie les remplacements de valeurs au moment du déploiement dans le manifeste de l'application. L'utilisation d'un fichier de remplacement vous permet d'utiliser le même manifeste d'application pour déployer l'application avec différents flux de caméras et configurer d'autres paramètres spécifiques à l'application.

Pour plus d'informations et des exemples de scripts pour chacune des étapes de cette rubrique, consultezAutomatisez le déploiement des applications.

# Gestion des applications dans la console AWS Panorama

Utilisez la console AWS Panorama pour gérer les applications déployées.

Sections



## Mettre à jour ou copier une application

Pour mettre à jour une application, utilisezRemplaceroption. Lorsque vous remplacez une application, vous pouvez mettre à jour son code ou ses modèles.

Pour mettre à jour une application

1. Ouvrez la console AWS PanoramaPage des applications déployées.
2. Choisissez une application.
3. Choisissez Remplacer.
4. Suivez les instructions pour créer une nouvelle version ou application.

Il y a également unCloneoption qui agit de la même manière queRemplacer, mais ne supprime pas l'ancienne version de l'application. Vous pouvez utiliser cette option pour tester les modifications apportées à une application sans arrêter la version en cours d'exécution, ou pour redéployer une version que vous avez déjà supprimée.

## Supprimer des versions et des applications

Pour nettoyer les versions d'applications inutilisées, supprimez-les de vos appliances.

Pour supprimer une application

1. Ouvrez la console AWS PanoramaPage des applications déployées.
2. Choisissez une application.
3. ChoisissezSuppression de l'appareil.

# Configuration de package

Lorsque vous utilisez la commande AWS Panorama Application CLI`panorama-cli package-application`, l'interface de ligne de commande télécharge les ressources de votre application sur Amazon S3 et les enregistre auprès d'AWS Panorama. Les ressources comprennent des fichiers binaires (images et modèles de conteneurs) et des fichiers descripteurs, téléchargés par AWS Panorama Appliance pendant le déploiement. Pour enregistrer les ressources d'un package, vous fournissez un fichier de configuration de package distinct qui définit le package, ses ressources et son interface.

L'exemple suivant montre une configuration de package pour un nœud de code avec une entrée et une sortie. L'entrée vidéo permet d'accéder aux données d'image à partir d'un flux de caméra. Le nœud de sortie envoie les images traitées vers un écran.

Example Paquets/1234567890-sample_code-1.0/package.json

```
{
    "nodePackage": {
        "envelopeVersion": "2021-01-01",
        "name": "SAMPLE_CODE",
        "version": "1.0",
        "description": "Computer vision application code.",
        "assets": [
            {
                "name": "code_asset",
                "implementations": [
                    {
                        "type": "container",
                        "assetUri":
 "3d9bxmplbdb67a3c9730abb19e48d78780b507f3340ec3871201903d8805328a.tar.gz",
                        "descriptorUri":
 "1872xmpl129481ed053c52e66d6af8b030f9eb69b1168a29012f01c7034d7a8f.json"
                    }
                ]
            }
        ],
        "interfaces": [
            {
                "name": "interface",
                "category": "business_logic",
                "asset": "code_asset",
                "inputs": [
```

```
                    {
                        "name": "video_in",
                        "type": "media"
                    }
                ],
                "outputs": [
                    {
                        "description": "Video stream output",
                        "name": "video_out",
                        "type": "media"
                    }
                ]
            }
        ]
    }
}
```

Le`assets`spécifie les noms des artefacts que l'interface de ligne de commande AWS Panorama Application a téléchargés sur Amazon S3. Si vous importez un exemple d'application ou une application provenant d'un autre utilisateur, cette section peut être vide ou faire référence à des ressources qui ne se trouvent pas dans votre compte. Lorsque vous courez`panorama-cli package-application`, l'AWS Panorama Application CLI remplit cette section avec les valeurs correctes.

# Le manifeste de l'application AWS Panorama

Lorsque vous déployez une application, vous fournissez un fichier de configuration appelé manifeste d'application. Ce fichier définit l'application comme un graphique avec des nœuds et des arêtes. Le manifeste de l'application fait partie du code source de l'application et est stocké dans le`graphs`répertoire.

Example Graphiquesaws-panorama-sample/graph.json

```
{
    "nodeGraph": {
        "envelopeVersion": "2021-01-01",
        "packages": [
            {
                "name": "123456789012::SAMPLE_CODE",
                "version": "1.0"
            },
            {
                "name": "123456789012::SQUEEZENET_PYTORCH_V1",
                "version": "1.0"
            },
            {
                "name": "panorama::abstract_rtsp_media_source",
                "version": "1.0"
            },
            {
                "name": "panorama::hdmi_data_sink",
                "version": "1.0"
            }
        ],
        "nodes": [
            {
                "name": "code_node",
                "interface": "123456789012::SAMPLE_CODE.interface"
            }
            {
                "name": "model_node",
                "interface": "123456789012::SQUEEZENET_PYTORCH_V1.interface"
            },
            {
                "name": "camera_node",
                "interface": "panorama::abstract_rtsp_media_source.rtsp_v1_interface",
                "overridable": true,
```

```
                "overrideMandatory": true,
                "decorator": {
                    "title": "IP camera",
                    "description": "Choose a camera stream."
                }
            },
            {
                "name": "output_node",
                "interface": "panorama::hdmi_data_sink.hdmi0"
            },
            {
                "name": "log_level",
                "interface": "string",
                "value": "INFO",
                "overridable": true,
                "decorator": {
                    "title": "Logging level",
                    "description": "DEBUG, INFO, WARNING, ERROR, or CRITICAL."
                }
            }
            ...
        ],
        "edges": [
            {
                "producer": "camera_node.video_out",
                "consumer": "code_node.video_in"
            },
            {
                "producer": "code_node.video_out",
                "consumer": "output_node.video_in"
            },
            {
                "producer": "log_level",
                "consumer": "code_node.log_level"
            }
        ]
    }
}
```

Les nœuds sont connectés par des tronçons, qui spécifient les mappages entre les entrées et les sorties des nœuds. La sortie d'un nœud se connecte à l'entrée d'un autre, formant un graphique.

# JSON

Le format des documents de manifeste et de remplacement de l'application est défini dans un schéma JSON. Vous pouvez utiliser le schéma JSON pour valider vos documents de configuration avant de les déployer. Le schéma JSON est disponible dans ce guideGitHubrepository.

- JSON–aws-panorama-developer-guide/ressources

# nœuds d'application

Les nœuds sont des modèles, du code, des flux de caméras, des sorties et des paramètres. Un nœud possède une interface qui définit ses entrées et ses sorties. L'interface peut être définie dans un package de votre compte, dans un package fourni par AWS Panorama ou dans un type intégré.

Dans l'exemple suivant,`code_node`et`model_node`Reportez-vous à l'exemple de code et de modèles inclus dans l'exemple d'application.`camera_node`utilise un package fourni par AWS Panorama pour créer un espace réservé pour un flux de caméra que vous spécifiez pendant le déploiement.

Example graph.json — Nœuds

```
        "nodes": [
            {
                "name": "code_node",
                "interface": "123456789012::SAMPLE_CODE.interface"
            },
            {
                "name": "model_node",
                "interface": "123456789012::SQUEEZENET_PYTORCH_V1.interface"
            },
            {
                "name": "camera_node",
                "interface": "panorama::abstract_rtsp_media_source.rtsp_v1_interface",
                "overridable": true,
                "overrideMandatory": true,
                "decorator": {
                    "title": "IP camera",
                    "description": "Choose a camera stream."
                }
            }
    ]
```

# Edges

Les bords mappent la sortie d'un nœud à l'entrée d'un autre. Dans l'exemple suivant, le premier Edge mappe la sortie d'un nœud de flux de caméra à l'entrée d'un nœud de code d'application. Les noms`video_in`et`video_out`sont définis dans les interfaces des paquetages de nœuds.

Example graph.json — bords

```
        "edges": [
```

```
        {
            "producer": "camera_node.video_out",
            "consumer": "code_node.video_in"
        },
        {
            "producer": "code_node.video_out",
            "consumer": "output_node.video_in"
        },
```

Dans votre code d'application, vous utilisez le`inputs`et`outputs`pour obtenir des images du flux d'entrée et envoyer des images au flux de sortie.

Example application.py — Entrée et sortie vidéo

```
    def process_streams(self):
        """Processes one frame of video from one or more video streams."""
        frame_start = time.time()
        self.frame_num += 1
        logger.debug(self.frame_num)
        # Loop through attached video streams
        streams = self.inputs.video_in.get()
        for stream in streams:
            self.process_media(stream)
        ...
        self.outputs.video_out.put(streams)
```

## Noeuds ab

Dans un manifeste d'application, un nœud abstrait fait référence à un package défini par AWS Panorama, que vous pouvez utiliser comme espace réservé dans votre manifeste d'application. AWS Panorama offre deux types de nœuds abstraits.

- flux de caméra: choisissez le flux de caméra utilisé par l'application pendant le déploiement.

  Nom du package–`panorama::abstract_rtsp_media_source`

  Nom d'interface–`rtsp_v1_interface`
- Sortie HDMI— Indique que l'application produit une vidéo.

  Nom du package–`panorama::hdmi_data_sink`

Nom d'interface–`hdmi0`

L'exemple suivant illustre un ensemble de packages, de nœuds et de tronçons de base pour une application qui traite les flux de caméra et qui produit des vidéos sur un écran. Le nœud de la caméra, qui utilise l'interface du`abstract_rtsp_media_source`dans AWS Panorama, peut accepter plusieurs flux de caméras en entrée. Le nœud de sortie, qui fait référence`hdmi_data_sink`, donne accès au code de l'application à une mémoire tampon vidéo sortie par le port HDMI de la solution matérielle-logicielle.

Example graph.json — Noeuds abstraits

```
{
    "nodeGraph": {
        "envelopeVersion": "2021-01-01",
        "packages": [
            {
                "name": "123456789012::SAMPLE_CODE",
                "version": "1.0"
            },
            {
                "name": "123456789012::SQUEEZENET_PYTORCH_V1",
                "version": "1.0"
            },
            {
                "name": "panorama::abstract_rtsp_media_source",
                "version": "1.0"
            },
            {
                "name": "panorama::hdmi_data_sink",
                "version": "1.0"
            }
        ],
        "nodes": [
            {
                "name": "camera_node",
                "interface": "panorama::abstract_rtsp_media_source.rtsp_v1_interface",
                "overridable": true,
                "decorator": {
                    "title": "IP camera",
                    "description": "Choose a camera stream."
                }
```

```
            },
            {
                "name": "output_node",
                "interface": "panorama::hdmi_data_sink.hdmi0"
            }
        ],
        "edges": [
            {
                "producer": "camera_node.video_out",
                "consumer": "code_node.video_in"
            },
            {
                "producer": "code_node.video_out",
                "consumer": "output_node.video_in"
            }
        ]
    }
}
```

# Paramètres de l'application

Les paramètres sont des nœuds de type basique et qui peuvent être remplacés pendant le déploiement. Un paramètre peut avoir une valeur par défaut et un paramètredécorateur, qui indique à l'utilisateur de l'application comment la configurer.

Types de paramètres

- `string`— une chaîne. Par exemple, `DEBUG`.
- `int32`— Un entier. Par exemple, `20`
- `float32`— Un nombre à virgule flottante. Par exemple, `47.5`
- `boolean`–`true`ou`false`.

L'exemple suivant montre deux paramètres, une chaîne et un nombre, qui sont envoyés à un nœud de code en entrée.

Example graph.json — Paramètres

```
       "nodes": [
           {
               "name": "detection_threshold",
               "interface": "float32",
               "value": 20.0,
               "overridable": true,
               "decorator": {
                   "title": "Threshold",
                   "description": "The minimum confidence percentage for a positive
classification."
               }
           },
           {
               "name": "log_level",
               "interface": "string",
               "value": "INFO",
               "overridable": true,
               "decorator": {
                   "title": "Logging level",
                   "description": "DEBUG, INFO, WARNING, ERROR, or CRITICAL."
               }
```

```
            }
            ...
        ],
        "edges": [
            {
                "producer": "detection_threshold",
                "consumer": "code_node.threshold"
            },
            {
                "producer": "log_level",
                "consumer": "code_node.log_level"
            }
            ...
        ]
    }
```

Vous pouvez modifier les paramètres directement dans le manifeste de l'application ou fournir de nouvelles valeurs au moment du déploiement avec des remplacements. Pour de plus amples informations, veuillez consulter Configuration au moment du déploiement avec remplacements.

# Configuration au moment du déploiement avec remplacements

Vous configurez les paramètres et les nœuds abstraits pendant le déploiement. Si vous utilisez la console AWS Panorama pour déployer, vous pouvez spécifier une valeur pour chaque paramètre et choisir un flux de caméra en entrée. Si vous utilisez l'API AWS Panorama pour déployer des applications, vous spécifiez ces paramètres avec un document de remplacement.

La structure d'un document de remplacement est similaire à celle d'un manifeste d'application. Pour les paramètres avec des types de base, vous définissez un nœud. Pour les flux de caméras, vous définissez un nœud et un package mappés à un flux de caméra enregistré. Vous définissez ensuite un remplacement pour chaque nœud qui spécifie le nœud à partir du manifeste d'application qu'il remplace.

Example overrides.json

```
{
    "nodeGraphOverrides": {
        "nodes": [
            {
                "name": "my_camera",
                "interface": "123456789012::exterior-south.exterior-south"
            },
            {
                "name": "my_region",
                "interface": "string",
                "value": "us-east-1"
            }
        ],
        "packages": [
            {
                "name": "123456789012::exterior-south",
                "version": "1.0"
            }
        ],
        "nodeOverrides": [
            {
                "replace": "camera_node",
                "with": [
                    {
                        "name": "my_camera"
                    }
                ]
```

```
            },
            {
                "replace": "region",
                "with": [
                    {
                        "name": "my_region"
                    }
                ]
            }
        ],
        "envelopeVersion": "2021-01-01"
    }
}
```

Dans l'exemple précédent, le document définit des remplacements pour un paramètre de chaîne et un nœud de caméra abstrait. Le`node0verrides`indique à AWS Panorama quels nœuds de ce document remplacent ceux dans le manifeste de l'application.

# Création d'applications AWS Panorama

Les applications exécutées surAWS PanoramaAppareil permettant d'effectuer des tâches de vision par ordinateur sur des flux vidéo. Vous pouvez créer des applications de vision par ordinateur en combinant du code Python et des modèles d'apprentissage automatique, et les déployer dans leAWS PanoramaAppareil appareil via Internet. Les applications peuvent envoyer une vidéo à un écran ou utiliser le kit SDK AWS pour envoyer des résultats aux services AWS.

UNmodèleanalyse des images pour détecter des personnes, des véhicules et d'autres objets. Sur la base des images qu'il a vues pendant l'entraînement, le modèle vous indique ce qu'il pense de quelque chose et à quel point il est confiant dans sa supposition. Vous pouvez entraîner des modèles avec vos propres données d'image ou commencer à utiliser un échantillon.

Le dossier de candidaturecodetraite des images fixes à partir d'un flux de caméra, les envoie à un modèle et traite le résultat. Un modèle peut détecter plusieurs objets et renvoyer leurs formes et leur emplacement. Le code peut utiliser ces informations pour ajouter du texte ou des graphiques à la vidéo, ou pour envoyer des résultats à unAWSservice de stockage ou de traitement ultérieur.

Pour obtenir des images à partir d'un flux, interagir avec un modèle et générer une vidéo, le code d'application utiliseleAWS PanoramaKit SDK pour applications. Le SDK de l'application est une bibliothèque Python qui prend en charge les modèles générés avec PyTorch, Apache MXNet et TensorFlow.

Rubriques

- Modèles de vision par ordinateur
- Création d'une image d'application
- Appeler les services AWS à partir du code de votre application
- Le kit SDK pour applications AWS Panorama
- Exécution de plusieurs threads
- Diffusion du trafic entrant
- Utilisation du GPU
- Configuration d'un environnement de développement sous Windows

# Modèles de vision par ordinateur

Un modèle de vision par ordinateur est un logiciel conçu pour détecter des objets dans des images. Un modèle apprend à reconnaître un ensemble d'objets en analysant d'abord des images de ces objets par le biais d'un entraînement. Un modèle de vision par ordinateur prend une image en entrée et génère des informations sur les objets qu'il détecte, tels que le type d'objet et son emplacement. AWS Panorama prend en charge les modèles de vision par ordinateur PyTorch créés avec Apache MXNet etTensorFlow.

> **Note**
>
> Pour obtenir la liste des modèles prédéfinis qui ont été testés avec AWS Panorama, consultez la section Compatibilité des modèles.

Sections



## Utilisation de modèles dans le code

Un modèle renvoie un ou plusieurs résultats, qui peuvent inclure des probabilités pour les classes détectées, des informations de localisation et d'autres données. L'exemple suivant montre comment exécuter une inférence sur une image à partir d'un flux vidéo et envoyer la sortie du modèle à une fonction de traitement.

Example application.py — Inférence

```
    def process_media(self, stream):
        """Runs inference on a frame of video."""
        image_data = preprocess(stream.image,self.MODEL_DIM)
        logger.debug('Image data: {}'.format(image_data))
        # Run inference
        inference_start = time.time()
        inference_results = self.call({"data":image_data}, self.MODEL_NODE)
         # Log metrics
```

```
        inference_time = (time.time() - inference_start) * 1000
        if inference_time > self.inference_time_max:
            self.inference_time_max = inference_time
        self.inference_time_ms += inference_time
        # Process results (classification)
        self.process_results(inference_results, stream)
```

L'exemple suivant montre une fonction qui traite les résultats du modèle de classification de base. Le modèle d'échantillon renvoie un tableau de probabilités, qui est la première et unique valeur du tableau de résultats.

Example application.py — Traitement des résultats

```
   def process_results(self, inference_results, stream):
       """Processes output tensors from a computer vision model and annotates a video
frame."""
       if inference_results is None:
           logger.warning("Inference results are None.")
           return
       max_results = 5
       logger.debug('Inference results: {}'.format(inference_results))
       class_tuple = inference_results[0]
       enum_vals = [(i, val) for i, val in enumerate(class_tuple[0])]
       sorted_vals = sorted(enum_vals, key=lambda tup: tup[1])
       top_k = sorted_vals[::-1][:max_results]
       indexes =  [tup[0] for tup in top_k]

       for j in range(max_results):
           label = 'Class [%s], with probability %.3f.'% (self.classes[indexes[j]],
class_tuple[0][indexes[j]])
           stream.add_label(label, 0.1, 0.1 + 0.1*j)
```

Le code de l'application trouve les valeurs présentant les probabilités les plus élevées et les mappe aux étiquettes d'un fichier de ressources chargé lors de l'initialisation.

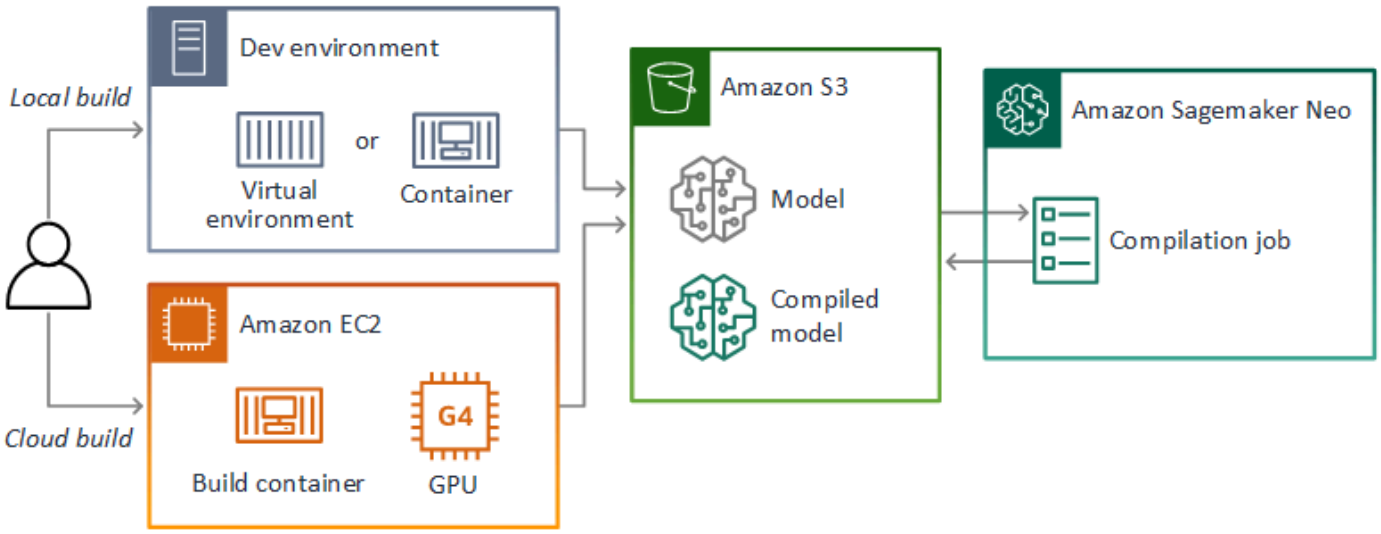
## Création d'un modèle personnalisé

Vous pouvez utiliser des modèles que vous créez dans PyTorch Apache MXNet et TensorFlow dans les applications AWS Panorama. Au lieu de créer et d'entraîner des modèles dansSageMaker, vous pouvez utiliser un modèle entraîné ou créer et entraîner votre propre modèle à l'aide d'un framework compatible et l'exporter dans un environnement local ou dans Amazon EC2.

> **Note**
>
> Pour en savoir plus sur les versions du framework et les formats de fichiers pris en charge par SageMaker Neo, consultez la section Frameworks pris en charge dans le manuel Amazon SageMaker Developer Guide.

Le référentiel de ce guide fournit un exemple d'application qui illustre ce flux de travail pour un modèle Keras au TensorFlow `SavedModel` format. Il utilise TensorFlow 2 et peut s'exécuter localement dans un environnement virtuel ou dans un conteneur Docker. L'exemple d'application inclut également des modèles et des scripts permettant de créer le modèle sur une instance Amazon EC2.

- Exemple d'application de modèle personnalisé




AWS Panorama utilise SageMaker Neo pour compiler des modèles à utiliser sur l'appliance AWS Panorama. Pour chaque framework, utilisez le format pris en charge par SageMaker Neo et empaquetez le modèle dans une `.tar.gz` archive.

Pour plus d'informations, consultez Compiler et déployer des modèles avec Neo dans le manuel Amazon SageMaker Developer Guide.

# Empaqueter un modèle

Un package de modèle comprend un descripteur, une configuration de package et une archive de modèles. Comme dans un package d'images d'application, la configuration du package indique au service AWS Panorama où le modèle et le descripteur sont stockés dans Amazon S3.

Example /Paquets/123456789012-squeezenet_pytorch-1.0/descriptor.json

```
{
    "mlModelDescriptor": {
        "envelopeVersion": "2021-01-01",
        "framework": "PYTORCH",
        "frameworkVersion": "1.8",
        "precisionMode": "FP16",
        "inputs": [
            {
                "name": "data",
                "shape": [
                    1,
                    3,
                    224,
                    224
                ]
            }
        ]
    }
}
```

> **Note**
>
> Spécifiez uniquement les versions majeure et secondaire de la version du framework. Pour obtenir la liste des versions et des TensorFlow versions d'Apache MXNet prises en chargePyTorch, consultez la section Frameworks pris en charge.

Pour importer un modèle, utilisez la `import-raw-model` commande CLI de l'application AWS Panorama. Si vous apportez des modifications au modèle ou à son descripteur, vous devez réexécuter cette commande pour mettre à jour les ressources de l'application. Pour plus d'informations, veuillez consulter Changer le modèle de vision par ordinateur.

Pour le schéma JSON du fichier descripteur, voir AssetDescriptor.schema.json.

# Entraînement de modèles

Lorsque vous entraînez un modèle, utilisez des images provenant de l'environnement cible ou d'un environnement de test qui ressemble étroitement à l'environnement cible. Tenez compte des facteurs suivants qui peuvent affecter les performances du modèle :

- Éclairage : la quantité de lumière réfléchie par un sujet détermine le niveau de détail que le modèle doit analyser. Un modèle entraîné avec des images de sujets bien éclairés peut ne pas fonctionner correctement dans un environnement faiblement éclairé ou à contre-jour.
- Résolution : la taille d'entrée d'un modèle est généralement fixée à une résolution comprise entre 224 et 512 pixels de large dans un format carré. Avant de transmettre une image vidéo au modèle, vous pouvez la réduire ou la recadrer pour l'adapter à la taille requise.
- Distorsion de l'image : la distance focale et la forme de l'objectif d'un appareil photo peuvent entraîner une distorsion des images loin du centre du cadre. La position d'une caméra détermine également quelles caractéristiques d'un sujet sont visibles. Par exemple, une caméra aérienne équipée d'un objectif grand angle affichera le haut d'un sujet lorsque celui-ci se trouve au centre du cadre, et une vue oblique du côté du sujet lorsqu'il s'éloigne du centre.

Pour résoudre ces problèmes, vous pouvez prétraiter les images avant de les envoyer au modèle et entraîner le modèle sur une plus grande variété d'images qui reflètent les variations dans des environnements réels. Si un modèle doit fonctionner dans des conditions d'éclairage et avec diverses caméras, vous avez besoin de plus de données pour l'entraînement. En plus de collecter davantage d'images, vous pouvez obtenir davantage de données d'entraînement en créant des variations de vos images existantes qui sont asymétriques ou présentent un éclairage différent.

# Création d'une image d'application

L'appliance AWS Panorama exécute les applications sous forme de systèmes de fichiers conteneurs exportés à partir d'une image que vous créez. Vous spécifiez les dépendances et les ressources de votre application dans un Dockerfile qui utilise l'image de base de l'application AWS Panorama comme point de départ.

Pour créer une image d'application, vous utilisez Docker et l'interface de ligne de commande AWS Panorama Application. L'exemple suivant, tiré de l'exemple d'application de ce guide, illustre ces cas d'utilisation.

Example /Paquets/123456789012-sample_code-1.0/Dockerfile

```
FROM public.ecr.aws/panorama/panorama-application
WORKDIR /panorama
COPY . .
RUN pip install --no-cache-dir --upgrade pip && \
    pip install --no-cache-dir -r requirements.txt
```

Les instructions Dockerfile suivantes sont utilisées.

- `FROM`— Charge l'image de base de l'application (`public.ecr.aws/panorama/panorama-application`).
- `WORKDIR`— Définit le répertoire de travail sur l'image `/panorama`est utilisé pour le code de l'application et les fichiers associés. Ce paramètre persiste uniquement pendant la génération et n'affecte pas le répertoire de travail de votre application lors de l'exécution (`/`).
- `COPY`— Copie les fichiers d'un chemin local vers un chemin de l'image. `COPY . .`copie les fichiers du répertoire actuel (le répertoire du package) vers le répertoire de travail de l'image. Par exemple, le code de l'application est copié de `packages/123456789012-SAMPLE_CODE-1.0/application.py` vers`/panorama/application.py`.
- `RUN`— Exécute des commandes shell sur l'image pendant la génération. Une seule `RUN` opération peut exécuter plusieurs commandes en séquence en les utilisant && entre les commandes. Cet exemple met à jour le gestionnaire de `pip` packages, puis installe les bibliothèques répertoriées dans`requirements.txt`.

Vous pouvez utiliser d'autres instructions, telles que `ADD` et`ARG`, qui s'avèrent utiles au moment de la génération. Les instructions qui ajoutent des informations d'exécution au conteneur, par exemple`ENV`,

ne fonctionnent pas avec AWS Panorama. AWS Panorama n'exécute pas de conteneur à partir de l'image. Il utilise uniquement l'image pour exporter un système de fichiers, qui est transféré vers l'appliance.

## Spécification des dépendances

`requirements.txt`est un fichier d'exigences Python qui spécifie les bibliothèques utilisées par l'application. L'exemple d'application utilise Open CV etAWS SDK for Python (Boto3).

Example /Paquets/123456789012-sample_code-1.0/requirements.txt

```
boto3==1.24.*
opencv-python==4.6.*
```

La `pip install` commande du Dockerfile installe ces bibliothèques dans le `dist-packages` répertoire Python ci-dessous`/usr/local/lib`, afin qu'elles puissent être importées par le code de votre application.

## Stockage local

AWS Panorama réserve le `/opt/aws/panorama/storage` répertoire pour le stockage des applications. Votre application peut créer et modifier des fichiers via ce chemin. Les fichiers créés dans le répertoire de stockage persistent après les redémarrages. Les autres emplacements de fichiers temporaires sont effacés au démarrage.

## Création de ressources d'image

Lorsque vous créez une image pour votre package d'application à l'aide de l'interface de ligne de commande AWS Panorama Application, cette dernière s'exécute `docker build` dans le répertoire du package. Cela permet de créer une image d'application qui contient le code de votre application. La CLI crée ensuite un conteneur, exporte son système de fichiers, le compresse et le stocke dans le `assets` dossier.

```
$ panorama-cli build-container --container-asset-name code_asset --package-path
 packages/123456789012-SAMPLE_CODE-1.0
docker build -t code_asset packages/123456789012-SAMPLE_CODE-1.0 --pull
docker export --output=code_asset.tar $(docker create code_asset:latest)
gzip -1 code_asset.tar
{
    "name": "code_asset",
```

```
    "implementations": [
        {
            "type": "container",
            "assetUri":
 "6f67xmpl32743ed0e60c151a02f2f0da1bf70a4ab9d83fe236fa32a6f9b9f808.tar.gz",
            "descriptorUri":
 "1872xmpl129481ed053c52e66d6af8b030f9eb69b1168a29012f01c7034d7a8f.json"
        }
    ]
}
Container asset for the package has been succesfully built at  /home/
user/aws-panorama-developer-guide/sample-apps/aws-panorama-sample/
assets/6f67xmpl32743ed0e60c151a02f2f0da1bf70a4ab9d83fe236fa32a6f9b9f808.tar.gz
```

Le bloc JSON dans la sortie est une définition d'actif que la CLI ajoute à la configuration du package (`package.json`) et enregistre auprès du service AWS Panorama. La CLI copie également le fichier descripteur, qui spécifie le chemin d'accès au script de l'application (le point d'entrée de l'application).

Example /Paquets/123456789012-sample_code-1.0/descriptor.json

```
{
    "runtimeDescriptor":
    {
        "envelopeVersion": "2021-01-01",
        "entry":
        {
            "path": "python3",
            "name": "/panorama/application.py"
        }
    }
}
```

Dans le dossier des ressources, le descripteur et l'image de l'application sont nommés d'après leur somme de contrôle SHA-256. Ce nom est utilisé comme identifiant unique pour l'actif lorsqu'il est stocké dans Amazon S3.

# Appeler les services AWS à partir du code de votre application

Vous pouvez utiliser leAWS SDK for Python (Boto) pour appeler les services AWS à partir du code de votre application. Par exemple, si votre modèle détecte quelque chose qui sort de l'ordinaire, vous pouvez publier des statistiques sur Amazon CloudWatch, envoyer une notification avec Amazon SNS, enregistrer une image sur Amazon S3 ou invoquer une fonction Lambda pour un traitement ultérieur. La plupart des services AWS disposent d'une API publique que vous pouvez utiliser avec le SDK AWS.

Par défaut, l'appliance n'est pas autorisée à accéder à des services AWS. Pour lui accorder l'autorisation, créez un rôle pour l'application et attribuez-le à l'instance de l'application lors du déploiement.

Sections



## Utilisation d'Amazon S3

Vous pouvez utiliser Amazon S3 pour stocker des résultats de traitement et d'autres données d'application.

```
import boto3
s3_client=boto3.client("s3")
s3_clients3.upload_file(data_file,
                    s3_bucket_name,
                    os.path.basename(data_file))
```

## Utilisation de la rubriqueAWS IoT MQTT

Vous pouvez utiliser le SDK for Python (Boto3) pour envoyer des messages à une rubrique MQTT dansAWS IoT. Dans l'exemple suivant, l'application publie dans une rubrique nommée d'après le nom de l'objet de l'appliance, que vous pouvez trouver dans AWS IoTla console.

```
import boto3
iot_client=boto3.client('iot-data')
topic = "panorama/panorama_my-appliance_Thing_a01e373b"
iot_client.publish(topic=topic, payload="my message")
```

Choisissez un nom qui indique l'identifiant de l'appareil ou un autre identifiant de votre choix. Pour publier des messages, l'application doit être autorisée à appeler`iot:Publish`.

Pour surveiller une file d'attente MQTT

1. Ouvrez la page de test de laAWS IoT console.
2. Dans Rubrique d'abonnement, saisissez le nom de la rubrique. Par exemple, `panorama/panorama_my-appliance_Thing_a01e373b`.
3. Choisissez Subscribe to topic (S'abonner à la rubrique).

# Le kit SDK pour applications AWS Panorama

Le kit SDK pour applications AWS Panorama est une bibliothèque Python permettant de développer des applications AWS Panorama. Dans voscode d'application, vous utilisez le kit SDK AWS Panorama Application pour charger un modèle de vision par ordinateur, exécuter une inférence et générer des vidéos sur un moniteur.

> **Note**
>
> Pour vous assurer d'avoir accès aux nouvelles fonctionnalités du kit SDK AWS Panorama,mettre à niveau le logiciel de l'appliance.

Pour plus d'informations sur les classes définies par le SDK d'application et leurs méthodes, reportez-vous à la sectionRéférence SDK des applications.

Sections

- Ajout de texte et de zones à la sortie vidéo

## Ajout de texte et de zones à la sortie vidéo

Avec le kit SDK AWS Panorama, vous pouvez générer un flux vidéo sur un écran. La vidéo peut inclure du texte et des zones affichant la sortie du modèle, l'état actuel de l'application ou d'autres données.

Chaque objet du`video_in`array est une image provenant d'un flux de caméra connecté à la solution matérielle-logicielle. Le type de cet objet est`panoramasdk.media`. Il dispose de méthodes permettant d'ajouter du texte et des zones rectangulaires à l'image, que vous pouvez ensuite attribuer à la`video_out`tableau.

Dans l'exemple suivant, l'exemple d'application ajoute une étiquette pour chacun des résultats. Chaque résultat est positionné à la même position gauche, mais à des hauteurs différentes.

```
        for j in range(max_results):
            label = 'Class [%s], with probability %.3f.'% (self.classes[indexes[j]],
class_tuple[0][indexes[j]])
            stream.add_label(label, 0.1, 0.1 + 0.1*j)
```

Pour ajouter une zone à l'image de sortie, utilisez`add_rect`. Cette méthode prend 4 valeurs comprises entre 0 et 1, ce qui indique la position des coins supérieur gauche et inférieur droit de la boîte.

```
        w,h,c = stream.image.shape
        stream.add_rect(x1/w, y1/h, x2/w, y2/h)
```

# Exécution de plusieurs threads

Vous pouvez exécuter la logique de votre application sur un thread de traitement et utiliser d'autres threads pour d'autres processus en arrière-plan. Par exemple, vous pouvez créer un threadsert le trafic HTTPpour le débogage, ou un thread qui surveille les résultats d'inférence et envoie des données àAWS.

Pour exécuter plusieurs threads, vous utilisez la commandeModule de threadsde la bibliothèque standard Python pour créer un thread pour chaque processus. L'exemple suivant montre la boucle principale de l'exemple d'application de serveur de débogage, qui crée un objet d'application et l'utilise pour exécutertroisthreads.

Example /Paquets/123456789012-debug_server-1.0/application.py— Boucle principale

```
def main():
    panorama = panoramasdk.node()
    while True:
        try:
            # Instantiate application
            logger.info('INITIALIZING APPLICATION')
            app = Application(panorama)
            # Create threads for stream processing, debugger, and client
            app.run_thread = threading.Thread(target=app.run_cv)
            app.server_thread = threading.Thread(target=app.run_debugger)
            app.client_thread = threading.Thread(target=app.run_client)
            # Start threads
            logger.info('RUNNING APPLICATION')
            app.run_thread.start()
            logger.info('RUNNING SERVER')
            app.server_thread.start()
            logger.info('RUNNING CLIENT')
            app.client_thread.start()
            # Wait for threads to exit
            app.run_thread.join()
            app.server_thread.join()
            app.client_thread.join()
            logger.info('RESTARTING APPLICATION')
        except:
            logger.exception('Exception during processing loop.')
```

Lorsque tous les threads se terminent, l'application redémarre d'elle-même. Lerun_cvLoop traite les images issues des flux de caméras. S'il reçoit un signal d'arrêt, il arrête le processus de

débogage, qui exécute un serveur HTTP et ne peut pas s'arrêter lui-même. Chaque thread doit gérer ses propres erreurs. Si une erreur n'est pas détectée et enregistrée, le thread se ferme en mode silencieux.

Example /Paquets/123456789012-debug_server-1.0/application.py— Boucle de traitement

```
    # Processing loop
    def run_cv(self):
        """Run computer vision workflow in a loop."""
        logger.info("PROCESSING STREAMS")
        while not self.terminate:
            try:
                self.process_streams()
                # turn off debug logging after 15 loops
                if logger.getEffectiveLevel() == logging.DEBUG and self.frame_num ==
15:
                    logger.setLevel(logging.INFO)
            except:
                logger.exception('Exception on processing thread.')
        # Stop signal received
        logger.info("SHUTTING DOWN SERVER")
        self.server.shutdown()
        self.server.server_close()
        logger.info("EXITING RUN THREAD")
```

Les threads communiquent via le`self`Objet Pour redémarrer la boucle de traitement de l'application, le thread de débogage appelle le`stop`Méthode. Cette méthode définit un`terminate`, qui indique aux autres threads de s'arrêter.

Example /Paquets/123456789012-debug_server-1.0/application.py— Méthode Stop

```
    # Interrupt processing loop
    def stop(self):
        """Signal application to stop processing."""
        logger.info("STOPPING APPLICATION")
        # Signal processes to stop
        self.terminate = True
    # HTTP debug server
    def run_debugger(self):
        """Process debug commands from local network."""
        class ServerHandler(SimpleHTTPRequestHandler):
            # Store reference to application
```

```
            application = self
            # Get status
            def do_GET(self):
                """Process GET requests."""
                logger.info('Get request to {}'.format(self.path))
                if self.path == "/status":
                    self.send_200('OK')
                else:
                    self.send_error(400)
            # Restart application
            def do_POST(self):
                """Process POST requests."""
                logger.info('Post request to {}'.format(self.path))
                if self.path == '/restart':
                    self.send_200('OK')
                    ServerHandler.application.stop()
                else:
                    self.send_error(400)
```

# Diffusion du trafic entrant

Vous pouvez surveiller ou déboguer des applications localement en exécutant un serveur HTTP en même temps que le code de votre application. Pour traiter le trafic externe, vous mappez les ports de l'appliance AWS Panorama aux ports de votre conteneur d'applications.

> **Important**
>
> Par défaut, l'appliance AWS Panorama n'accepte le trafic entrant sur aucun port. L'ouverture de ports sur l'appliance présente un risque de sécurité implicite. Lorsque vous utilisez cette fonction, vous devez prendre des mesures supplémentaires poursécurisez votre appliance contre le trafic externeet des communications sécurisées entre les clients autorisés et l'appliance.
> L'exemple de code inclus dans ce guide est fourni à des fins de démonstration et n'implémente pas l'authentification, l'autorisation ou le chiffrement.

Vous pouvez ouvrir des ports compris entre 8000 et 9000 sur l'appliance. Ces ports, lorsqu'ils sont ouverts, peuvent recevoir du trafic provenant de n'importe quel client routable. Lorsque vous déployez votre application, vous spécifiez les ports à ouvrir et vous mappez les ports de l'appliance aux ports de votre conteneur d'applications. Le logiciel de l'appliance transmet le trafic au conteneur et renvoie les réponses au demandeur. Les demandes sont reçues sur le port de l'appliance que vous spécifiez et les réponses sont envoyées sur un port éphémère aléatoire.

## Configuration des ports entrants

Vous spécifiez des mappages de ports à trois endroits dans la configuration de votre application. Le package de code`package.json`, vous spécifiez le port sur lequel le nœud de code écoute dans un`network`bloquer. L'exemple suivant déclare que le nœud écoute sur le port 80.

Example /Paquets/123456789012-debug_server-1.0/package.json

```
            "outputs": [
                {
                    "description": "Video stream output",
                    "name": "video_out",
                    "type": "media"
                }
```

```
            ],
            "network": {
                "inboundPorts": [
                    {
                        "port": 80,
                        "description": "http"
                    }
                ]
            }
```

Dans le manifeste de l'application, vous déclarez une règle de routage qui mappe un port de l'appliance à un port du conteneur de code de l'application. L'exemple suivant ajoute une règle qui mappe le port 8080 de l'appareil au port 80 du`code_node`conteneur.

Example graphs/mon-app/graph.json

```
        {
            "producer": "model_input_width",
            "consumer": "code_node.model_input_width"
        },
        {
            "producer": "model_input_order",
            "consumer": "code_node.model_input_order"
        }
    ],
    "networkRoutingRules": [
        {
            "node": "code_node",
            "containerPort": 80,
            "hostPort": 8080,
            "decorator": {
                "title": "Listener port 8080",
                "description": "Container monitoring and debug."
            }
        }
    ]
```

Lorsque vous déployez l'application, vous spécifiez les mêmes règles dans la console AWS Panorama ou avec un document de remplacement transmis auCreateApplicationInstanceAPI. Vous devez fournir cette configuration au moment du déploiement pour confirmer que vous souhaitez ouvrir des ports sur l'appliance.

Example graphiques/mon-app/override.json

```
            {
                "replace": "camera_node",
                "with": [
                    {
                        "name": "exterior-north"
                    }
                ]
            }
        ],
        "networkRoutingRules":[
            {
                "node": "code_node",
                "containerPort": 80,
                "hostPort": 8080
            }
        ],
        "envelopeVersion": "2021-01-01"
    }
}
```

Si le port de périphérique spécifié dans le manifeste de l'application est utilisé par une autre application, vous pouvez utiliser le document de remplacement pour choisir un autre port.

## Diffusion du trafic

Lorsque les ports sont ouverts sur le conteneur, vous pouvez ouvrir un socket ou exécuter un serveur pour gérer les requêtes entrantes. Le`debug-server`L'exemple montre une implémentation de base d'un serveur HTTP exécuté parallèlement au code d'application de vision par ordinateur.

> **Important**
>
> L'exemple d'implémentation n'est pas sécurisé pour une utilisation en production. Pour éviter de rendre votre appliance vulnérable aux attaques, vous devez implémenter des contrôles de sécurité appropriés dans votre code et votre configuration réseau.

Example /Paquets/123456789012-debug_server-1.0/application.py— Serveur HTTP

```
    # HTTP debug server
```

```
    def run_debugger(self):
        """Process debug commands from local network."""
        class ServerHandler(SimpleHTTPRequestHandler):
            # Store reference to application
            application = self
            # Get status
            def do_GET(self):
                """Process GET requests."""
                logger.info('Get request to {}'.format(self.path))
                if self.path == '/status':
                    self.send_200('OK')
                else:
                    self.send_error(400)
            # Restart application
            def do_POST(self):
                """Process POST requests."""
                logger.info('Post request to {}'.format(self.path))
                if self.path == '/restart':
                    self.send_200('OK')
                    ServerHandler.application.stop()
                else:
                    self.send_error(400)
            # Send response
            def send_200(self, msg):
                """Send 200 (success) response with message."""
                self.send_response(200)
                self.send_header('Content-Type', 'text/plain')
                self.end_headers()
                self.wfile.write(msg.encode('utf-8'))
        try:
            # Run HTTP server
            self.server = HTTPServer(("", self.CONTAINER_PORT), ServerHandler)
            self.server.serve_forever(1)
            # Server shut down by run_cv loop
            logger.info("EXITING SERVER THREAD")
        except:
            logger.exception('Exception on server thread.')
```

Le serveur accepte les requêtes GET au niveau du`/status`chemin pour récupérer certaines informations sur l'application. Il accepte également une demande POST vers`/restart`pour redémarrer l'application.

Pour démontrer cette fonctionnalité, l'exemple d'application exécute un client HTTP sur un thread distinct. Le client appelle le`/status`chemin sur le réseau local peu de temps après le démarrage et redémarre l'application quelques minutes plus tard.

Example /Paquets/123456789012-debug_server-1.0/application.py— Client HTTP

```
    # HTTP test client
    def run_client(self):
        """Send HTTP requests to device port to demnostrate debug server functions."""
        def client_get():
            """Get container status"""
            r = requests.get('http://{}:{}/status'.format(self.device_ip,
 self.DEVICE_PORT))
            logger.info('Response: {}'.format(r.text))
            return
        def client_post():
            """Restart application"""
            r = requests.post('http://{}:{}/restart'.format(self.device_ip,
 self.DEVICE_PORT))
            logger.info('Response: {}'.format(r.text))
            return
        # Call debug server
        while not self.terminate:
            try:
                time.sleep(30)
                client_get()
                time.sleep(300)
                client_post()
            except:
                logger.exception('Exception on client thread.')
        # stop signal received
        logger.info("EXITING CLIENT THREAD")
```

La boucle principale gère les threads et redémarre l'application lorsqu'ils quittent.

Example /Paquets/123456789012-debug_server-1.0/application.py— Boucle principale

```
def main():
    panorama = panoramasdk.node()
    while True:
        try:
            # Instantiate application
            logger.info('INITIALIZING APPLICATION')
```

```
            app = Application(panorama)
            # Create threads for stream processing, debugger, and client
            app.run_thread = threading.Thread(target=app.run_cv)
            app.server_thread = threading.Thread(target=app.run_debugger)
            app.client_thread = threading.Thread(target=app.run_client)
            # Start threads
            logger.info('RUNNING APPLICATION')
            app.run_thread.start()
            logger.info('RUNNING SERVER')
            app.server_thread.start()
            logger.info('RUNNING CLIENT')
            app.client_thread.start()
            # Wait for threads to exit
            app.run_thread.join()
            app.server_thread.join()
            app.client_thread.join()
            logger.info('RESTARTING APPLICATION')
        except:
            logger.exception('Exception during processing loop.')
```

Pour déployer l'exemple d'application, consultezinstructions de ce guide GitHub repository.

# Utilisation du GPU

Vous pouvez accéder au processeur graphique (GPU) de l'appliance AWS Panorama pour utiliser des bibliothèques accélérées par GPU ou exécuter des modèles d'apprentissage automatique dans le code de votre application. Pour activer l'accès au GPU, vous devez obligatoirement ajouter l'accès au GPU à la configuration du package après avoir créé votre conteneur de code d'application.

> **Important**
>
> Si vous activez l'accès au GPU, vous ne pouvez pas exécuter de nœuds modèles dans aucune application de l'appliance. Pour des raisons de sécurité, l'accès au GPU est restreint lorsque l'appliance exécute un modèle compilé avec SageMaker Neo. Avec l'accès au GPU, vous devez exécuter vos modèles dans des nœuds de code d'application, et toutes les applications de l'appareil partagent l'accès au GPU.

Pour activer l'accès au GPU pour votre application, mettez à jour la configuration du package après avoir créé le package à l'aide de l'interface de ligne de commande AWS Panorama Application. L'exemple suivant montre le `requirements` bloc qui ajoute un accès GPU au nœud de code de l'application.

Example package.json avec bloc d'exigences

```
{
    "nodePackage": {
        "envelopeVersion": "2021-01-01",
        "name": "SAMPLE_CODE",
        "version": "1.0",
        "description": "Computer vision application code.",
        "assets": [
            {
                "name": "code_asset",
                "implementations": [
                    {
                        "type": "container",
                        "assetUri":
 "eba3xmpl71aa387e8f89be9a8c396416cdb80a717bb32103c957a8bf41440b12.tar.gz",
                        "descriptorUri":
 "4abdxmpl5a6f047d2b3047adde44704759d13f0126c00ed9b4309726f6bb43400ba9.json",
                        "requirements": [
```

```
                        {
                            "type": "hardware_access",
                            "inferenceAccelerators": [
                                {
                                    "deviceType": "nvhost_gpu",
                                    "sharedResourcePolicy": {
                                        "policy" : "allow_all"
                                    }
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "interfaces": [
    ...
```

Mettez à jour la configuration du package entre les étapes de création et d'empaquetage de votre flux de travail de développement.

Pour déployer une application avec accès au GPU

1. Pour créer le conteneur d'applications, utilisez la `build-container` commande.

   ```
   $ panorama-cli build-container --container-asset-name code_asset --package-path
    packages/123456789012-SAMPLE_CODE-1.0
   ```

2. Ajoutez le `requirements` bloc à la configuration du package.
3. Pour charger la configuration de la ressource et du package du conteneur, utilisez la `package-application` commande.

   ```
   $ panorama-cli package-application
   ```

4. Déployez l'application.

Pour des exemples d'applications utilisant l'accès au GPU, visitez le aws-panorama-samplesGitHubréférentiel.

# Configuration d'un environnement de développement sous Windows

Pour créer une application AWS Panorama, vous utilisez Docker, des outils de ligne de commande et Python. Sous Windows, vous pouvez configurer un environnement de développement à l'aide de Docker Desktop avec Windows Subsystem for Linux et Ubuntu. Ce didacticiel vous guide tout au long du processus de configuration d'un environnement de développement testé avec des outils AWS Panorama et des exemples d'applications.

Sections



## Prérequis

Pour suivre ce didacticiel, vous avez besoin d'une version de Windows prenant en charge le sous-système Windows pour Linux 2 (WSL 2).

- Windows 10 version 1903 et supérieure (version 18362 et supérieure) ou Windows 11
- Fonctionnalités Windows
  - WSL (Windows Subsystem for Linux)
  - Hyper-V
  - Plateforme de machines virtuelles

Ce didacticiel a été développé avec les versions logicielles suivantes.

- Ubuntu 20.04
- Python 3.8.5
- Docker 20.10.8

## Installez WSL 2 et Ubuntu

Si vous disposez de Windows 10 version 2004 et supérieure (version 19041 et supérieure), vous pouvez installer WSL 2 et Ubuntu 20.04 à l'aide de la commande PowerShell suivante.

```
> wsl --install -d Ubuntu-20.04
```

Pour une ancienne version de Windows, suivez les instructions fournies dans la documentation WSL 2 : Étapes manuelles pour les anciennes versions

## Installer Docker

Pour installer Docker Desktop, téléchargez et exécutez le package d'installation à partir de partir dehub.docker.com. Si vous rencontrez des problèmes, suivez les instructions sur le site Web Docker : Backend Docker Desktop WSL 2.

Exécutez Docker Desktop et suivez le didacticiel de première exécution pour créer un exemple de conteneur.

> **Note**
>
> Docker Desktop n'active Docker que dans la distribution par défaut. Si d'autres distributions Linux sont installées avant d'exécuter ce didacticiel, activez Docker dans la distribution Ubuntu nouvellement installée dans le menu des paramètres Docker Desktop sousRessources,Intégration WSL.

## Configurer Ubuntu

Vous pouvez désormais exécuter des commandes Docker sur votre machine virtuelle Ubuntu. Pour ouvrir un terminal de ligne de commande, exécutez la distribution à partir du menu Démarrer. La première fois que vous l'exécutez, vous configurez un nom d'utilisateur et un mot de passe que vous pouvez utiliser pour exécuter des commandes d'administrateur.

Pour terminer la configuration de votre environnement de développement, mettez à jour le logiciel de la machine virtuelle et installez les outils.

Pour configurer la machine virtuelle

1. Mettez à jour le logiciel fourni avec Ubuntu.

```
$ sudo apt update && sudo apt upgrade -y && sudo apt autoremove
```

2. Installez des outils de développement avec apt.

```
$ sudo apt install unzip python3-pip
```

3. Installez les bibliothèques Python avec pip.

```
$ pip3 install awscli panoramacli
```

4. Ouvrez un nouveau terminal, puis exécutez`aws configure`pour configurer leAWS CLI.

```
$ aws configure
```

   Si vous n'avez pas de clés d'accès, vous pouvez les générer dans laConsole IAM.

Enfin, téléchargez et importez l'exemple d'application.

Pour obtenir l'exemple d'application

1. Téléchargez et extrayez l'exemple d'application.

```
$ wget https://github.com/awsdocs/aws-panorama-developer-guide/releases/download/
v1.0-ga/aws-panorama-sample.zip
$ unzip aws-panorama-sample.zip
$ cd aws-panorama-sample
```

2. Exécutez les scripts inclus pour tester la compilation, créer le conteneur d'applications et télécharger des packages sur AWS Panorama.

```
aws-panorama-sample$ ./0-test-compile.sh
aws-panorama-sample$ ./1-create-role.sh
aws-panorama-sample$ ./2-import-app.sh
aws-panorama-sample$ ./3-build-container.sh
aws-panorama-sample$ ./4-package-app.sh
```

L'AWS Panorama Application CLI télécharge des packages et les enregistre auprès du service AWS Panorama. Vous pouvez maintenantDéploiement de l'exemple d'applicationavec la console AWS Panorama.

## Étapes suivantes

Pour explorer et modifier les fichiers du projet, vous pouvez utiliser l'Explorateur de fichiers ou un environnement de développement intégré (IDE) prenant en charge WSL.

Pour accéder au système de fichiers de la machine virtuelle, ouvrez l'Explorateur de fichiers et entrez`\\wsl$`dans la barre de navigation. Ce répertoire contient un lien vers le système de fichiers de la machine virtuelle (`Ubuntu-20.04`) et des systèmes de fichiers pour les données Docker. Under`Ubuntu-20.04`, votre annuaire d'utilisateurs se trouve à`home\`*`username`*.

> **Note**
>
> Pour accéder aux fichiers de votre installation Windows à partir d'Ubuntu, accédez au`/mnt/`crépertoire. Par exemple, vous pouvez lister les fichiers de votre répertoire de téléchargements en exécutant`ls /mnt/c/Users/`*`windows-username`*`/Downloads`.

Avec Visual Studio Code, vous pouvez modifier le code d'application dans votre environnement de développement et exécuter des commandes avec un terminal intégré. Pour installer Visual Studio Code, rendez-vous surcode.visualstudio.com. Après l'installation, ajoutez leWSL à distanceextension.

Le terminal Windows est une alternative au terminal Ubuntu standard dans lequel vous exécutez des commandes. Il prend en charge plusieurs onglets et peut exécuter PowerShell, l'invite de commandes et les terminaux pour n'importe quelle autre variété de Linux que vous installez. Il prend en charge le copier-coller avecCtrl+CetCtrl+V, des URL cliquables et d'autres améliorations utiles. Pour installer Windows Terminal, rendez-vous surmicrosoft.com.

# L'API AWS Panorama

Vous pouvez utiliser l'API publique du service AWS Panorama pour automatiser les flux de travail de gestion des appareils et des applications. Avec le SDKAWS Command Line Interface ou leAWS SDK, vous pouvez développer des scripts ou des applications qui gèrent les ressources et les déploiements. Ce guide inclut GitHub des scripts que vous pouvez utiliser comme point de départ pour votre propre code.

- aws-panorama-developer-guide/scripts utilitaires

Sections

# Automatisez l'enregistrement des appareils

Pour provisionner une appliance, utilisez leProvisionDeviceAPI. La réponse inclut un fichier ZIP avec la configuration de l'appareil et les informations d'identification temporaires. Décodez le fichier et enregistrez-le dans une archive avec le préfixe`certificates-omni_`.

Example provision-device.sh

```
if [[ $# -eq 1 ]] ; then
    DEVICE_NAME=$1
else
    echo "Usage: ./provision-device.sh <device-name>"
    exit 1
fi
CERTIFICATE_BUNDLE=certificates-omni_${DEVICE_NAME}.zip
aws panorama provision-device --name ${DEVICE_NAME} --output text --query Certificates
 | base64 --decode > ${CERTIFICATE_BUNDLE}
echo "Created certificate bundle ${CERTIFICATE_BUNDLE}"
```

Les informations d'identification de l'archive de configuration expirent au bout de 5 minutes. Transférez l'archive vers votre appliance à l'aide de la clé USB incluse.

Pour enregistrer une caméra, utilisez leCréer un nœud à partir d'une tâche de modèleAPI. Cette API prend une carte des paramètres de modèle pour le nom d'utilisateur, le mot de passe et l'URL de la caméra. Vous pouvez formater cette carte en tant que document JSON à l'aide de la manipulation de chaînes Bash.

Example register-camera.sh

```
if [[ $# -eq 3 ]] ; then
    NAME=$1
    USERNAME=$2
    URL=$3
else
    echo "Usage: ./register-camera.sh <stream-name> <username> <rtsp-url>"
    exit 1
fi
echo "Enter camera stream password: "
read PASSWORD
TEMPLATE='{"Username":"MY_USERNAME","Password":"MY_PASSWORD","StreamUrl": "MY_URL"}'
TEMPLATE=${TEMPLATE/MY_USERNAME/$USERNAME}
```

```
TEMPLATE=${TEMPLATE/MY_PASSWORD/$PASSWORD}
TEMPLATE=${TEMPLATE/MY_URL/$URL}
echo ${TEMPLATE}
JOB_ID=$(aws panorama create-node-from-template-job --template-type RTSP_CAMERA_STREAM
 --output-package-name ${NAME} --output-package-version "1.0" --node-name ${NAME} --
template-parameters "${TEMPLATE}" --output text)
```

Vous pouvez également charger la configuration JSON à partir d'un fichier.

```
--template-parameters file://camera-template.json
```

# Gérez les appliances avec l'API AWS Panorama

Vous pouvez automatiser les tâches de gestion des appliances grâce à l'API AWS Panorama.

## Afficher les appareils

Pour obtenir la liste des appliances avec des identifiants d'appareils, utilisez l'ListDevicesAPI.

```
$ aws panorama list-devices
    "Devices": [
        {
            "DeviceId": "device-4tafxmplhtmzabv5lsacba4ere",
            "Name": "my-appliance",
            "CreatedTime": 1652409973.613,
            "ProvisioningStatus": "SUCCEEDED",
            "LastUpdatedTime": 1652410973.052,
            "LeaseExpirationTime": 1652842940.0
        }
    ]
}
```

Pour obtenir plus de détails sur une appliance, utilisez l'DescribeDeviceAPI.

```
$ aws panorama describe-device --device-id device-4tafxmplhtmzabv5lsacba4ere
{
    "DeviceId": "device-4tafxmplhtmzabv5lsacba4ere",
    "Name": "my-appliance",
    "Arn": "arn:aws:panorama:us-west-2:123456789012:device/
device-4tafxmplhtmzabv5lsacba4ere",
    "Type": "PANORAMA_APPLIANCE",
    "DeviceConnectionStatus": "ONLINE",
    "CreatedTime": 1648232043.421,
    "ProvisioningStatus": "SUCCEEDED",
    "LatestSoftware": "4.3.55",
    "CurrentSoftware": "4.3.45",
    "SerialNumber": "GFXMPL0013023708",
    "Tags": {},
    "CurrentNetworkingStatus": {
        "Ethernet0Status": {
            "IpAddress": "192.168.0.1/24",
            "ConnectionStatus": "CONNECTED",
            "HwAddress": "8C:XM:PL:60:C5:88"
        },
```

```
        "Ethernet1Status": {
            "IpAddress": "--",
            "ConnectionStatus": "NOT_CONNECTED",
            "HwAddress": "8C:XM:PL:60:C5:89"
        }
    },
    "LeaseExpirationTime": 1652746098.0
}
```

## Mettre à niveau le logiciel du dispositif

Si la`LatestSoftware` version est plus récente que la`CurrentSoftware`, vous pouvez mettre à niveau l'appareil. Utilisez l'CreateJobForDevicesAPI pour créer une tâche de mise à jour over-the-air (OTA).

```
$ aws panorama create-job-for-devices --device-ids device-4tafxmplhtmzabv5lsacba4ere \
  --device-job-config '{"OTAJobConfig": {"ImageVersion": "4.3.55"}}' --job-type OTA
{
    "Jobs": [
        {
            "JobId": "device-4tafxmplhtmzabv5lsacba4ere-0",
            "DeviceId": "device-4tafxmplhtmzabv5lsacba4ere"
        }
    ]
}
```

Dans un script, vous pouvez renseigner le champ de version de l'image dans le fichier de configuration de la tâche avec une manipulation de chaîne Bash.

Example check-updates.sh

```
apply_update() {
    DEVICE_ID=$1
    NEW_VERSION=$2
    CONFIG='{"OTAJobConfig": {"ImageVersion": "NEW_VERSION"}}'
    CONFIG=${CONFIG/NEW_VERSION/$NEW_VERSION}
    aws panorama create-job-for-devices --device-ids ${DEVICE_ID} --device-job-config
 "${CONFIG}" --job-type OTA
}
```

L'appliance télécharge la version logicielle spécifiée et se met à jour elle-même. Suivez la progression de la mise à jour avec l'DescribeDeviceJobAPI.

```
$ aws panorama describe-device-job --job-id device-4tafxmplhtmzabv5lsacba4ere-0
{
    "JobId": "device-4tafxmplhtmzabv5lsacba4ere-0",
    "DeviceId": "device-4tafxmplhtmzabv5lsacba4ere",
    "DeviceArn": "arn:aws:panorama:us-west-2:559823168634:device/
device-4tafxmplhtmzabv5lsacba4ere",
    "DeviceName": "my-appliance",
    "DeviceType": "PANORAMA_APPLIANCE",
    "ImageVersion": "4.3.55",
    "Status": "REBOOTING",
    "CreatedTime": 1652410232.465
}
```

Pour obtenir la liste de toutes les tâches en cours d'exécution, utilisez le ListDevicesJobs.

```
$ aws panorama list-devices-jobs
{
    "DeviceJobs": [
        {
            "DeviceName": "my-appliance",
            "DeviceId": "device-4tafxmplhtmzabv5lsacba4ere",
            "JobId": "device-4tafxmplhtmzabv5lsacba4ere-0",
            "CreatedTime": 1652410232.465
        }
    ]
}
```

Pour obtenir un exemple de script qui vérifie et applique les mises à jour, consultez le fichier check-updates.sh dans le GitHub référentiel de ce guide.

## Redémarrer les dispositifs

Pour redémarrer une appliance, utilisez l'CreateJobForDevicesAPI.

```
$ aws panorama create-job-for-devices --device-ids device-4tafxmplhtmzabv5lsacba4ere --
job-type REBOOT
{
    "Jobs": [
        {
            "JobId": "device-4tafxmplhtmzabv5lsacba4ere-0",
            "DeviceId": "device-4tafxmplhtmzabv5lsacba4ere"
        }
```

```
    ]
}
```

Dans un script, vous pouvez obtenir la liste des appareils et en choisir un à redémarrer de manière interactive.

Example reboot-device.sh — utilisation

```
$ ./reboot-device.sh
Getting devices...
0: device-53amxmplyn3gmj72epzanacniy     my-se70-1
1: device-6talxmpl5mmik6qh5moba6jium     my-manh-24
Choose a device
1
Reboot device device-6talxmpl5mmik6qh5moba6jium? (y/n)y
{
    "Jobs": [
        {
            "DeviceId": "device-6talxmpl5mmik6qh5moba6jium",
            "JobId": "device-6talxmpl5mmik6qh5moba6jium-8"
        }
    ]
}
```

# Automatisez le déploiement des applications

Pour déployer une application, vous utilisez à la fois l'interface de ligne de commande AWS Panorama Application etAWS Command Line Interface. Après avoir créé le conteneur d'applications, vous le chargez ainsi que d'autres actifs vers un point d'accès Amazon S3. Vous déployez ensuite l'application à l'aide duCreateApplicationInstanceAPI.

Pour plus de contexte et pour obtenir des instructions sur l'utilisation des scripts présentés, suivez les instructions duexemple d'application README.

Sections

- Construisez le conteneur
- Téléchargez le conteneur et enregistrez les nœuds
- Déployer l'application
- Surveiller le déploiement

## Construisez le conteneur

Pour créer le conteneur d'applications, utilisez`build-container`commande. Cette commande crée un conteneur Docker et l'enregistre sous forme de système de fichiers compressé dans`assets`dossier.

Example 3-build-container.sh

```
CODE_PACKAGE=SAMPLE_CODE
ACCOUNT_ID=$(aws sts get-caller-identity --output text --query 'Account')
panorama-cli build-container --container-asset-name code_asset --package-path packages/
${ACCOUNT_ID}-${CODE_PACKAGE}-1.0
```

Vous pouvez également utiliser la saisie automatique par ligne de commande pour renseigner l'argument path en saisissant une partie du chemin, puis en appuyant surTAB.

```
$ panorama-cli build-container --package-path packages/TAB
```

## Téléchargez le conteneur et enregistrez les nœuds

Pour charger l'application, utilisez`package-application`commande. Cette commande télécharge des ressources depuis`assets`dossier vers un point d'accès Amazon S3 géré par AWS Panorama.

Example 4-package-app.sh

```
panorama-cli package-application
```

L'interface de ligne de commande de l'application AWS Panorama télécharge les ressources du conteneur et du descripteur référencées par la configuration du package (`package.json`) dans chaque package, et enregistre les packages en tant que nœuds dans AWS Panorama. Vous faites ensuite référence à ces nœuds dans le manifeste de votre application (`graph.json`) pour déployer l'application.

## Déployer l'application

Pour déployer l'application, vous utilisezCreateApplicationInstanceAPI. Cette action prend notamment en compte les paramètres suivants.

- `ManifestPayload`— Le manifeste de l'application (`graph.json`) qui définit les nœuds, les packages, les arêtes et les paramètres de l'application.
- `ManifestOverridesPayload`— Un second manifeste qui remplace les paramètres du premier. Le manifeste de l'application peut être considéré comme une ressource statique dans la source de l'application, où le manifeste de remplacement fournit des paramètres de temps de déploiement qui personnalisent le déploiement.
- `DefaultRuntimeContextDevice`— L'appareil cible.
- `RuntimeRoleArn`— L'ARN d'un rôle IAM que l'application utilise pour accéder aux services et aux ressources AWS.
- `ApplicationInstanceIdToReplace`— L'ID d'une instance d'application existante à supprimer de l'appareil.

Le manifeste et les charges utiles de remplacement sont des documents JSON qui doivent être fournis sous la forme d'une valeur de chaîne imbriquée dans un autre document. Pour ce faire, le script charge les manifestes à partir d'un fichier sous forme de chaîne et utiliseoutil jqpour créer le document imbriqué.

Example 5-deploy.sh— composer des manifestes

```
GRAPH_PATH="graphs/my-app/graph.json"
OVERRIDE_PATH="graphs/my-app/override.json"
# application manifest
GRAPH=$(cat ${GRAPH_PATH} | tr -d '\n' | tr -d '[:blank:]')
MANIFEST="$(jq --arg value "${GRAPH}" '.PayloadData="\($value)"' <<< {})"
# manifest override
OVERRIDE=$(cat ${OVERRIDE_PATH} | tr -d '\n' | tr -d '[:blank:]')
MANIFEST_OVERRIDE="$(jq --arg value "${OVERRIDE}" '.PayloadData="\($value)"' <<< {})"
```

Le script de déploiement utiliseListDevicesAPI permettant d'obtenir une liste des appareils enregistrés dans la région actuelle et d'enregistrer le choix de l'utilisateur dans un fichier local pour les déploiements ultérieurs.

Example 5-deploy.sh— trouver un appareil

```
    echo "Getting devices..."
    DEVICES=$(aws panorama list-devices)
    DEVICE_NAMES=($((echo ${DEVICES} | jq -r '.Devices |=sort_by(.LastUpdatedTime) |
 [.Devices[].Name] | @sh') | tr -d \'\"))
    DEVICE_IDS=($((echo ${DEVICES} | jq -r '.Devices |=sort_by(.LastUpdatedTime) |
 [.Devices[].DeviceId] | @sh') | tr -d \'\"))
    for (( c=0; c<${#DEVICE_NAMES[@]}; c++ ))
    do
        echo "${c}: ${DEVICE_IDS[${c}]}     ${DEVICE_NAMES[${c}]}"
    done
    echo "Choose a device"
    read D_INDEX
    echo "Deploying to device ${DEVICE_IDS[${D_INDEX}]}"
    echo -n ${DEVICE_IDS[${D_INDEX}]} > device-id.txt
    DEVICE_ID=$(cat device-id.txt)
```

Le rôle d'application est créé par un autre script (1-create-role.sh). Le script de déploiement obtient l'ARN de ce rôle depuisAWS CloudFormation. Si l'application est déjà déployée sur l'appareil, le script obtient l'ID de cette instance d'application à partir d'un fichier local.

Example 5-deploy.sh— ARN de rôle et arguments de remplacement

```
# application role
STACK_NAME=panorama-${NAME}
ROLE_ARN=$(aws cloudformation describe-stacks --stack-name panorama-${PWD##*/} --query
 'Stacks[0].Outputs[?OutputKey==`roleArn`].OutputValue' --output text)
```

```
ROLE_ARG="--runtime-role-arn=${ROLE_ARN}"

# existing application instance id
if [ -f "application-id.txt" ]; then
    EXISTING_APPLICATION=$(cat application-id.txt)
    REPLACE_ARG="--application-instance-id-to-replace=${EXISTING_APPLICATION}"
    echo "Replacing application instance ${EXISTING_APPLICATION}"
fi
```

Enfin, le script réunit tous les éléments pour créer une instance d'application et déployer l'application sur l'appareil. Le service répond avec un ID d'instance que le script enregistre pour une utilisation ultérieure.

Example 5-deploy.sh— déployer l'application

```
APPLICATION_ID=$(aws panorama create-application-instance ${REPLACE_ARG} --manifest-
payload="${MANIFEST}" --default-runtime-context-device=${DEVICE_ID} --name=${NAME}
 --description="command-line deploy" --tags client=sample --manifest-overrides-
payload="${MANIFEST_OVERRIDE}" ${ROLE_ARG} --output text)
echo "New application instance ${APPLICATION_ID}"
echo -n $APPLICATION_ID > application-id.txt
```

## Surveiller le déploiement

Pour surveiller un déploiement, utilisezListApplicationInstancesAPI. Le script de surveillance obtient l'ID du périphérique et l'ID de l'instance d'application à partir des fichiers du répertoire de l'application et les utilise pour créer une commande CLI. Il appelle ensuite en boucle.

Example 6-monitor-deployment.sh

```
APPLICATION_ID=$(cat application-id.txt)
DEVICE_ID=$(cat device-id.txt)
QUERY="ApplicationInstances[?ApplicationInstanceId==\`APPLICATION_ID\`]"
QUERY=${QUERY/APPLICATION_ID/$APPLICATION_ID}
MONITOR_CMD="aws panorama list-application-instances --device-id ${DEVICE_ID} --query
 ${QUERY}"
MONITOR_CMD=${MONITOR_CMD/QUERY/$QUERY}
while true; do
    $MONITOR_CMD
    sleep 60
done
```

Une fois le déploiement terminé, vous pouvez consulter les journaux en appelant AmazonCloudWatchAPI de journalisation. Le script d'affichage des journaux utiliseCloudWatchJournauxAPI.`GetLogEvents`

Example view-logs.sh

```
GROUP="/aws/panorama/devices/MY_DEVICE_ID/applications/MY_APPLICATION_ID"
GROUP=${GROUP/MY_DEVICE_ID/$DEVICE_ID}
GROUP=${GROUP/MY_APPLICATION_ID/$APPLICATION_ID}
echo "Getting logs for group ${GROUP}."
#set -x
while true
do
    LOGS=$(aws logs get-log-events --log-group-name ${GROUP} --log-stream-name
 code_node --limit 150)
    readarray -t ENTRIES < <(echo $LOGS | jq -c '.events[].message')
    for ENTRY in "${ENTRIES[@]}"; do
        echo "$ENTRY" | tr -d \"
    done
    sleep 20
done
```

# Gérez les applications avec l'API AWS Panorama

Vous pouvez contrôlez et gérer des applications à l'aide d'AWS Panorama API.

## Vue en applications

Pour obtenir la liste des applications exécutées sur une appliance, utilisez l'ListApplicationInstancesAPI.

```
$ aws panorama list-application-instances
    "ApplicationInstances": [
        {
            "Name": "aws-panorama-sample",
            "ApplicationInstanceId": "applicationInstance-ddaxxmpl2z7bg74ywutd7byxuq",
            "DefaultRuntimeContextDevice": "device-4tafxmplhtmzabv5lsacba4ere",
            "DefaultRuntimeContextDeviceName": "my-appliance",
            "Description": "command-line deploy",
            "Status": "DEPLOYMENT_SUCCEEDED",
            "HealthStatus": "RUNNING",
            "StatusDescription": "Application deployed successfully.",
            "CreatedTime": 1661902051.925,
            "Arn": "arn:aws:panorama:us-east-2:123456789012:applicationInstance/
applicationInstance-ddaxxmpl2z7bg74ywutd7byxuq",
            "Tags": {
                "client": "sample"
            }
        },
    ]
}
```

Pour obtenir plus de détails sur les nœuds d'une instance d'application, utilisez l'ListApplicationInstanceNodeInstancesAPI.

```
$ aws panorama list-application-instance-node-instances --application-instance-id
 applicationInstance-ddaxxmpl2z7bg74ywutd7byxuq
{
    "NodeInstances": [
        {
            "NodeInstanceId": "code_node",
            "NodeId": "SAMPLE_CODE-1.0-fd3dxmpl-interface",
            "PackageName": "SAMPLE_CODE",
```

```
            "PackageVersion": "1.0",
            "PackagePatchVersion":
 "fd3dxmpl2bdfa41e6fe1be290a79dd2c29cf014eadf7416d861ce7715ad5e8a8",
            "NodeName": "interface",
            "CurrentStatus": "RUNNING"
        },
        {
            "NodeInstanceId": "camera_node_override",
            "NodeId": "warehouse-floor-1.0-9eabxmpl-warehouse-floor",
            "PackageName": "warehouse-floor",
            "PackageVersion": "1.0",
            "PackagePatchVersion":
 "9eabxmple89f0f8b2f2852cca2a6e7971aa38f1629a210d069045e83697e42a7",
            "NodeName": "warehouse-floor",
            "CurrentStatus": "RUNNING"
        },
        {
            "NodeInstanceId": "output_node",
            "NodeId": "hdmi_data_sink-1.0-9c23xmpl-hdmi0",
            "PackageName": "hdmi_data_sink",
            "PackageVersion": "1.0",
            "PackagePatchVersion":
 "9c23xmplc4c98b92baea4af676c8b16063d17945a3f6bd8f83f4ff5aa0d0b394",
            "NodeName": "hdmi0",
            "CurrentStatus": "RUNNING"
        },
        {
            "NodeInstanceId": "model_node",
            "NodeId": "SQUEEZENET_PYTORCH-1.0-5d3cabda-interface",
            "PackageName": "SQUEEZENET_PYTORCH",
            "PackageVersion": "1.0",
            "PackagePatchVersion":
 "5d3cxmplb7113faa1d130f97f619655d8ca12787c751851a0e155e50eb5e3e96",
            "NodeName": "interface",
            "CurrentStatus": "RUNNING"
        }
    ]
}
```

## Gestion des flux de caméras

Vous pouvez suspendre et reprendre les nœuds de flux de caméra à l'aide de l'SignalApplicationInstanceNodeInstancesAPI.

```
$ aws panorama signal-application-instance-node-instances --application-instance-id
 applicationInstance-ddaxxmpl2z7bg74ywutd7byxuq \
       --node-signals '[{"NodeInstanceId": "camera_node_override", "Signal":
 "PAUSE"}]'
{
    "ApplicationInstanceId": "applicationInstance-ddaxxmpl2z7bg74ywutd7byxuq"
}
```

Dans un script, vous pouvez obtenir une liste de nœuds et en choisir un à suspendre ou à reprendre de manière interactive.

Example pause-camera.sh — utilisation

```
my-app$ ./pause-camera.sh

Getting nodes...
0: SAMPLE_CODE              RUNNING
1: warehouse-floor          RUNNING
2: hdmi_data_sink           RUNNING
3: entrance-north           PAUSED
4: SQUEEZENET_PYTORCH       RUNNING
Choose a node
1
Signalling node warehouse-floor
+ aws panorama signal-application-instance-node-instances --application-instance-id
 applicationInstance-r3a7xmplcbmpjqeds7vj4b6pjy --node-signals '[{"NodeInstanceId":
 "warehouse-floor", "Signal": "PAUSE"}]'
{
    "ApplicationInstanceId": "applicationInstance-r3a7xmplcbmpjqeds7vj4b6pjy"
}
```

En interrompant et en reprenant les nœuds de caméra, vous pouvez parcourir un plus grand nombre de flux de caméra que ceux qui peuvent être traités simultanément. Pour ce faire, mappez plusieurs flux de caméras vers le même nœud d'entrée dans votre manifeste de remplacement.

Dans l'exemple suivant, le manifeste de remplacement mappe deux flux de caméra`entrance-north` vers le même nœud d'entrée (`camera_node`).`warehouse-floor` Le`warehouse-floor` flux est actif lorsque l'application démarre et que le`entrance-north` nœud attend qu'un signal soit activé.

Example override-multicam.json

```
    "nodeGraphOverrides": {
        "nodes": [
            {
                "name": "warehouse-floor",
                "interface": "123456789012::warehouse-floor.warehouse-floor",
                "launch": "onAppStart"
            },
            {
                "name": "entrance-north",
                "interface": "123456789012::entrance-north.entrance-north",
                "launch": "onSignal"
            },
        ...
        "packages": [
            {
                "name": "123456789012::warehouse-floor",
                "version": "1.0"
            },
            {
                "name": "123456789012::entrance-north",
                "version": "1.0"
            }
        ],
        "nodeOverrides": [
            {
                "replace": "camera_node",
                "with": [
                    {
                        "name": "warehouse-floor"
                    },
                    {
                        "name": "entrance-north"
                    }
                ]
            }
```

Pour plus de détails sur le déploiement avec l'API, consultezAutomatisez le déploiement des applications.

# Utilisation de points de terminaison d'un VPC

Si vous travaillez dans un VPC sans accès à Internet, vous pouvez créer un point de terminaison VPC à utiliser avec AWS Panorama. Un point de terminaison VPC permet aux clients s'exécutant dans un sous-réseau privé de se connecter à un service AWS sans connexion Internet.

Pour plus de détails sur les ports et les points de terminaison utilisés par l'appliance AWS Panorama, consultez???.

Sections

- Création d'un point de terminaison d'un VPC
- Connexion d'une appliance à un sous-réseau privé
- Exemples de AWS CloudFormation modèles

## Création d'un point de terminaison d'un VPC

Pour établir une connexion privée entre votre VPC et AWS Panorama, créez un point de terminaison VPC. Il n'est pas nécessaire de disposer d'un point de terminaison VPC pour utiliser AWS Panorama. Vous ne devez créer un point de terminaison VPC que si vous travaillez dans un VPC sans accès à Internet. Lorsque la CLI ou le SDK AWS tente de se connecter à AWS Panorama, le trafic est acheminé via le point de terminaison VPC.

Créez un point de terminaison VPC pour AWS Panorama à l'aide des paramètres suivants :

- Nom du service — **`com.amazonaws.`*`us-west-2`*`.panorama`**
- Type — Interface

Un point de terminaison VPC utilise le nom DNS du service pour obtenir le trafic des clients du SDK AWS sans aucune configuration supplémentaire. Pour plus d'informations sur l'utilisation des points de terminaison VPC, consultez la section Points de terminaison VPC d'interface dans le guide de l'utilisateur Amazon VPC.

## Connexion d'une appliance à un sous-réseau privé

L'appliance AWS Panorama peut se connecter AWS via une connexion VPN privée avec AWS Site-to-Site VPN ouAWS Direct Connect. Grâce à ces services, vous pouvez créer un sous-réseau privé

qui s'étend à votre centre de données. L'appliance se connecte au sous-réseau privé et accède aux AWS services via les points de terminaison VPC.

Les VPN de site à site AWS Direct Connect sont des services permettant de connecter votre centre de données à Amazon VPC en toute sécurité. Avec le VPN Site-to-Site, vous pouvez utiliser des appareils réseau disponibles dans le commerce pour vous connecter. AWS Direct Connectutilise un AWS appareil pour se connecter.

- VPN de site à site : qu'est-ce que c'est ? AWS Site-to-Site VPN
- AWS Direct Connect— Qu'est-ce que c'est AWS Direct Connect ?

Après avoir connecté votre réseau local à un sous-réseau privé d'un VPC, créez des points de terminaison VPC pour les services suivants.

- Amazon Simple Storage Service — AWS PrivateLinkpour Amazon S3
- AWS IoT Core— Utilisation AWS IoT Core avec les points de terminaison VPC de l'interface (plan de données et fournisseur d'informations d'identification)
- Amazon Elastic Container Registry — Points de terminaison VPC de l'interface Amazon Elastic Container Registry
- Amazon CloudWatch — Utilisation CloudWatch avec des points de terminaison VPC d'interface
- Amazon CloudWatch Logs — Utilisation des CloudWatch journaux avec des points de terminaison VPC d'interface

L'appliance n'a pas besoin d'être connectée au service AWS Panorama. Il communique avec AWS Panorama via un canal de messagerie enAWS IoT.

Outre les points de terminaison VPC, Amazon S3 AWS IoT nécessite l'utilisation de zones hébergées privées Amazon Route 53. La zone hébergée privée achemine le trafic depuis les sous-domaines, y compris les sous-domaines des points d'accès Amazon S3 et des sujets MQTT, vers le point de terminaison VPC approprié. Pour plus d'informations sur les zones hébergées privées, consultez la section Travailler avec des zones hébergées privées dans le guide du développeur Amazon Route 53.

Pour un exemple de configuration VPC avec des points de terminaison VPC et des zones hébergées privées, consultez. Exemples de AWS CloudFormation modèles

# Exemples de AWS CloudFormation modèles

Le GitHub référentiel de ce guide fournit des AWS CloudFormation modèles que vous pouvez utiliser pour créer des ressources à utiliser avec AWS Panorama. Les modèles créent un VPC avec deux sous-réseaux privés, un sous-réseau public et un point de terminaison VPC. Vous pouvez utiliser les sous-réseaux privés du VPC pour héberger des ressources isolées d'Internet. Les ressources du sous-réseau public peuvent communiquer avec les ressources privées, mais les ressources privées ne sont pas accessibles depuis Internet.

Example vpc-endpoint.yml — Sous-réseaux privés

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 172.31.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
      Tags:
        - Key: Name
          Value: !Ref AWS::StackName
  privateSubnetA:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref vpc
      AvailabilityZone:
        Fn::Select:
         - 0
         - Fn::GetAZs: ""
      CidrBlock: 172.31.3.0/24
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub  ${AWS::StackName}-subnet-a
  ...
```

Le `vpc-endpoint.yml` modèle montre comment créer un point de terminaison VPC pour AWS Panorama. Vous pouvez utiliser ce point de terminaison pour gérer les ressources AWS Panorama avec le AWS SDK ouAWS CLI.

Example vpc-endpoint.yml — Point de terminaison VPC

```
panoramaEndpoint:
  Type: AWS::EC2::VPCEndpoint
  Properties:
    ServiceName: !Sub com.amazonaws.${AWS::Region}.panorama
    VpcId: !Ref vpc
    VpcEndpointType: Interface
    SecurityGroupIds:
    - !GetAtt vpc.DefaultSecurityGroup
    PrivateDnsEnabled: true
    SubnetIds:
    - !Ref privateSubnetA
    - !Ref privateSubnetB
    PolicyDocument:
      Version: 2012-10-17
      Statement:
      - Effect: Allow
        Principal: "*"
        Action:
          - "panorama:*"
        Resource:
          - "*"
```

`PolicyDocument`Il s'agit d'une politique d'autorisation basée sur les ressources qui définit les appels d'API qui peuvent être effectués avec le point de terminaison. Vous pouvez modifier la politique afin de restreindre les actions et les ressources accessibles via le point de terminaison. Pour plus d'informations, consultez Contrôle de l'accès aux services avec points de terminaison d'un VPC dans le Guide de l'utilisateur Amazon VPC.

Le `vpc-appliance.yml` modèle montre comment créer des points de terminaison VPC et des zones hébergées privées pour les services utilisés par l'appliance AWS Panorama.

Example vpc-appliance.yml — Point de terminaison du point d'accès Amazon S3 avec zone hébergée privée

```
s3Endpoint:
  Type: AWS::EC2::VPCEndpoint
  Properties:
    ServiceName: !Sub com.amazonaws.${AWS::Region}.s3
    VpcId: !Ref vpc
    VpcEndpointType: Interface
```

```
      SecurityGroupIds:
      - !GetAtt vpc.DefaultSecurityGroup
      PrivateDnsEnabled: false
      SubnetIds:
      - !Ref privateSubnetA
      - !Ref privateSubnetB
...
  s3apHostedZone:
    Type: AWS::Route53::HostedZone
    Properties:
      Name: !Sub s3-accesspoint.${AWS::Region}.amazonaws.com
      VPCs:
        - VPCId: !Ref vpc
          VPCRegion: !Ref AWS::Region
  s3apRecords:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref s3apHostedZone
      Name: !Sub "*.s3-accesspoint.${AWS::Region}.amazonaws.com"
      Type: CNAME
      TTL: 600
      # first DNS entry, split on :, second value
      ResourceRecords:
      - !Select [1, !Split [":", !Select [0, !GetAtt s3Endpoint.DnsEntries ] ] ]
```

Les exemples de modèles illustrent la création de ressources Amazon VPC et Route 53 avec un exemple de VPC. Vous pouvez les adapter à votre cas d'utilisation en supprimant les ressources VPC et en remplaçant les références aux identifiants de sous-réseau, de groupe de sécurité et de VPC par les identifiants de vos ressources.

# Exemples d'applications, de scripts et de modèles

Le GitHub le référentiel de ce guide fournit des exemples d'applications, de scripts et de modèles pourAWS Panoramaappareils. Utilisez ces exemples pour découvrir les meilleures pratiques et automatiser les flux de travail de développement.

Sections



## Exemples d'applications

Des exemples d'applications démontrent l'utilisation deAWS Panoramafonctionnalités et tâches courantes de vision par ordinateur. Ces exemples d'applications incluent des scripts et des modèles qui automatisent la configuration et le déploiement. Avec une configuration minimale, vous pouvez déployer et mettre à jour des applications à partir de la ligne de commande.

- aws-panorama-sample— Vision par ordinateur de base avec un modèle de classification. Utilisez leAWS SDK for Python (Boto)pour télécharger des métriques vers CloudWatch, les méthodes de prétraitement et d'inférence des instruments, et configurez la journalisation.
- serveur de débogage—Ports entrants ouvertssur l'appareil et transférez le trafic vers un conteneur de code d'application. Utilisez le multithreading pour exécuter du code d'application, un serveur HTTP et un client HTTP simultanément.
- modèle personnalisé— Exportez des modèles depuis le code et compilez avec SageMaker Neo pour tester la compatibilité avecAWS PanoramaAppareil. Construisez localement dans un développement Python, dans un conteneur Docker ou sur une instance Amazon EC2. Exportez et compilez tous les modèles d'applications intégrés dans Keras pour une application spécifique TensorFlow ou version Python.

Pour d'autres exemples d'applications, consultez également leaws-panorama-samplesréférentiel.

# Scripts utilitaires

Les scripts du`util-scripts`gestion d'annuairesAWS Panoramaressources ou automatisez les flux de travail de développement.

- provision-device.sh— Fournir un appareil.
- check-updates.sh— Vérifiez les mises à jour logicielles de l'appliance et appliquez-les.
- reboot-device.sh— Redémarrez un appareil.
- register-camera.sh— Enregistrez une caméra.
- deregister-camera.sh— Supprime un nœud de caméra.
- view-logs.sh— Afficher les journaux d'une instance d'application.
- pause-camera.sh— Suspendez ou reprenez le flux d'une caméra.
- push.sh— Créez, téléchargez et déployez une application.
- rename-package.sh— Renomme un package de nœuds. Met à jour les noms de répertoire, les fichiers de configuration et le manifeste de l'application.
- samplify.sh— Remplacez votre identifiant de compte par un exemple d'identifiant de compte et restaurez les configurations de sauvegarde pour supprimer la configuration locale.
- update-model-config.sh— Ajoutez de nouveau le modèle à l'application après avoir mis à jour le fichier descripteur.
- cleanup-patches.sh— Désenregistrez les anciennes versions des correctifs et supprimez leurs manifestes d'Amazon S3.

Pour plus de détails sur l'utilisation, voirle README.

# AWS CloudFormation Modèles

Utilisez leAWS CloudFormationmodèles dans le`cloudformation-templates`répertoire pour créer des ressources pourAWS Panoramaapplications.

- alarm-application.yml— Créez une alarme qui surveille les erreurs d'une application. Si l'instance d'application génère des erreurs ou s'arrête de fonctionner pendant 5 minutes, l'alarme envoie un e-mail de notification.

- dispositif d'alarme.yml— Créez une alarme qui surveille la connectivité d'un appareil. Si l'appareil arrête d'envoyer des métriques pendant 5 minutes, l'alarme envoie un e-mail de notification.
- rôle-d'application .yml— Créez un rôle d'application. Le rôle inclut l'autorisation d'envoyer des métriques à CloudWatch. Ajoutez des autorisations à la déclaration de politique pour les autres opérations d'API utilisées par votre application.
- vpc-appliance.yml— Créez un VPC avec accès au service de sous-réseau privé pourAWS PanoramaAppareil. Pour connecter l'appliance à un VPC, utilisezAWS Direct ConnectouAWS Site-to-Site VPN.
- vpc-endpoint.yml— Créez un VPC avec un accès de service de sous-réseau privé auAWS Panoramaservice. Les ressources à l'intérieur du VPC peuvent se connecter àAWS Panoramapour surveiller et gérerAWS Panoramaressources sans connexion à Internet.

Le`create-stack.sh`un script dans ce répertoire créeAWS CloudFormationpiles. Elle prend un nombre variable d'arguments. Le premier argument est le nom du modèle, et les autres arguments sont des remplacements de paramètres du modèle.

Par exemple, la commande suivante crée un rôle d'application.

```
$ ./create-stack.sh application-role
```

## Plus d'échantillons et d'outils

Leaws-panorama-samplesle référentiel contient plus d'exemples d'applications et d'outils utiles.

- Demandes— Exemples d'applications pour différentes architectures de modèles et différents cas d'utilisation.
- Validation du flux de caméras— Validez les flux de caméras.
- PanoJupyter— Courir JupyterLab sur unAWS PanoramaAppareil.
- Chargement latéral— Mettez à jour le code de l'application sans créer ni déployer de conteneur d'applications.

LeAWSla communauté a également développé des outils et des conseils pourAWS Panorama. Découvrez les projets open source suivants sur GitHub.

- emporte-pièce - panorama— Un modèle Cookiecutter pourAWS Panoramaapplications.

- sac à dos— Modules Python permettant d'accéder aux détails de l'environnement d'exécution, au profilage et aux options de sortie vidéo supplémentaires.

# SurveillanceAWS Panoramaressources et applications

Vous pouvez surveillerAWS Panoramaressources dans leAWS Panoramaconsole et avec AmazonCloudWatch. LeAWS PanoramaL'appliance se connecte auAWSCloud sur Internet pour signaler son état et l'état des caméras connectées. Lorsqu'elle est sous tension, la solution matérielle-logicielle envoie également des journaux àCloudWatchJournaux en temps réel.

La solution matérielle-logicielle obtient l'autorisation d'utiliserAWS IoT,CloudWatchJournaux et autres services AWS à partir d'un rôle de service que vous créez la première fois que vous utilisez leAWS Panoramaconsole Pour plus d'informations, consultez Rôles de service et ressources interservices AWS Panorama.

Pour obtenir de l'aide sur le dépannage d'erreurs spécifiques, voirRésolution des problèmes.

Rubriques

- Surveillance dans la console AWS Panorama
- Affichage des journaux AWS Panorama
- Surveillance des appliances et des applications avec AmazonCloudWatch

# Surveillance dans la console AWS Panorama

Vous pouvez utiliser la console AWS Panorama pour surveiller votre appliance AWS Panorama et vos caméras. La console utiliseAWS IoTpour surveiller l'état de la solution matérielle-logicielle.

Pour surveiller votre appliance dans la console AWS Panorama

1. Ouverture d'Console AWS Panorama.
2. Ouvrez la console AWS PanoramaPage des appareils.
3. Choisissez une appliance.
4. Pour voir l'état d'une instance d'application, choisissez-la dans la liste.
5. Pour voir l'état des interfaces réseau de la solution matérielle-logicielle, choisissezParamètres.

L'état général de l'appliance s'affiche en haut de la page. Si le statut estEn ligne, puis la solution matérielle-logicielle est connectée àAWSet envoyer des mises à jour régulières d'état.

# Affichage des journaux AWS Panorama

AWS Panorama signale les événements relatifs aux applications et au système à Amazon CloudWatch Journaux. Lorsque vous rencontrez des problèmes, vous pouvez utiliser les journaux d'événements pour déboguer votre application AWS Panorama ou résoudre les problèmes de configuration de l'application.

Pour afficher les journaux CloudWatch Journaux

1. Ouvrir lePage Groupes de journaux de la CloudWatch Console Logs.
2. Trouvez les journaux des applications et des appliances AWS Panorama dans les groupes suivants :

   - Journaux des appareils–`/aws/panorama/devices/`*`device-id`*
   - Journaux d'applications–`/aws/panorama/devices/`*`device-id`*`/applications/`*`instance-id`*

Lorsque vous reprovisionnez une appliance après avoir mis à jour le logiciel système, vous pouvez égalementAffichage des journaux sur le lecteur USB de provisioning.

Sections

- Affichage des journaux d'appareils
- Affichage des journaux d'application
- Configuration des journaux d'application
- Affichage des journaux d'approvisionnement
- Extraction de journaux depuis un appareil

## Affichage des journaux d'appareils

L'appliance AWS Panorama crée un groupe de journaux pour l'appareil et un groupe pour chaque instance d'application que vous déployez. Les journaux des appareils contiennent des informations sur l'état de l'application, les mises à niveau logicielles et la configuration du système.

Journaux des appareils —**/aws/panorama/devices/*device-id***

- `occ_log`— Résultat du processus du contrôleur. Ce processus coordonne les déploiements d'applications et fournit des rapports sur l'état des nœuds de chaque instance d'application.
- `ota_log`— Résultat du processus qui coordonne over-the-air mises à niveau logicielles (OTA).
- `syslog`— Résultat du processus Syslog de l'appareil, qui capture les messages envoyés entre les processus.
- `kern_log`— Événements provenant du noyau Linux de l'appareil.
- `logging_setup_logs`— Résultat du processus qui configure le CloudWatch Agent Logs.
- `cloudwatch_agent_logs`— Données de sortie de CloudWatch Agent Logs.
- `shadow_log`— Données de sortie deAWS IoTombre de l'appareil.

## Affichage des journaux d'application

Le groupe de journaux d'une instance d'application contient un flux de journaux pour chacun des nœuds, nommé d'après le nœud.

Journaux d'applications —**/aws/panorama/devices/*device-id*/applications/*instance-id***

- Code— Sortie à partir du code de votre application et du SDK de l'application AWS Panorama. Regroupe les journaux d'applications provenant de`/opt/aws/panorama/logs`.
- Modèle— Résultat du processus qui coordonne les demandes d'inférence avec un modèle.
- Stream— Résultat du processus qui décode la vidéo d'un flux de caméra.
- Afficher— Sortie du processus qui effectue le rendu de la sortie vidéo pour le port HDMI.
- `mds`— Journaux provenant du serveur de métadonnées de l'appliance.
- `console_output`— Capture les flux d'erreurs et de sortie standard à partir des conteneurs de code

Si vous ne voyez pas les journaux CloudWatch Journaux, confirmez que vous êtes dans la région AWS correcte. Si tel est le cas, il peut exister un problème avec la connexion de l'appliance à AWS ou avec les autorisations surde l'applianceAWS Identity and Access Managementrôle (IAM).

# Configuration des journaux d'application

Configuration d'un enregistreur Python pour écrire des fichiers journaux`/opt/aws/panorama/logs`. L'appliance diffuse les journaux depuis cet emplacement vers CloudWatch Journaux. Pour éviter d'utiliser trop d'espace disque, utilisez une taille de fichier maximale de 10 MiB et un nombre de sauvegardes de 1. L'exemple suivant montre une méthode permettant de créer un enregistreur.

Example application.py— Configuration de l'enregistreur

```
def get_logger(name=__name__,level=logging.INFO):
    logger = logging.getLogger(name)
    logger.setLevel(level)
    LOG_PATH = '/opt/aws/panorama/logs'
    handler = RotatingFileHandler("{}/app.log".format(LOG_PATH), maxBytes=10000000,
 backupCount=1)
    formatter = logging.Formatter(fmt='%(asctime)s %(levelname)-8s %(message)s',
                                  datefmt='%Y-%m-%d %H:%M:%S')
    handler.setFormatter(formatter)
    logger.addHandler(handler)
    return logger
```

Initialisez l'enregistreur à portée globale et utilisez-le dans tout le code de votre application.

Example application.py— Initialize Dlogger

```
def main():
    try:
        logger.info("INITIALIZING APPLICATION")
        app = Application()
        logger.info("PROCESSING STREAMS")
        while True:
            app.process_streams()
            # turn off debug logging after 150 loops
            if logger.getEffectiveLevel() == logging.DEBUG and app.frame_num == 150:
                logger.setLevel(logging.INFO)
    except:
        logger.exception('Exception during processing loop.')

logger = get_logger(level=logging.INFO)
main()
```

# Affichage des journaux d'approvisionnement

Lors du provisionnement, l'appliance AWS Panorama copie les journaux sur la clé USB que vous utilisez pour transférer l'archive de configuration vers l'appliance. Utilisez ces journaux pour résoudre les problèmes de provisionnement sur les appliances dotées de la dernière version logicielle.

> **Important**
>
> Les journaux de provisionnement sont disponibles pour les appliances mises à jour vers la version logicielle 4.3.23 ou une version ultérieure.

Journaux d'application

- `/panorama/occ.log`— Journaux du logiciel AWS Panorama Controller.
- `/panorama/ota_agent.log`— AWS Panorama over-the-air mettre à jour les journaux des agents
- `/panorama/syslog.log`— Journaux du système Linux.
- `/panorama/kern.log`— Journaux du noyau Linux.

# Extraction de journaux depuis un appareil

Si les journaux de votre appareil et de vos applications n'apparaissent pas dans CloudWatch Journaux, vous pouvez utiliser une clé USB pour extraire une image de journal cryptée de l'appareil. L'équipe du service AWS Panorama peut déchiffrer les journaux en votre nom et vous aider à déboguer.

## Prérequis

Pour suivre la procédure, vous aurez besoin du matériel suivant :

- Clé USB— Une clé USB au format FAT32 avec au moins 1 Go de stockage, pour transférer les fichiers journaux depuis l'appliance AWS Panorama.

Pour extraire les journaux de l'appareil

1. Préparez une clé USB avec un`managed_logs`dossier à l'intérieur d'un`panorama`dossier.

```
/
 ### panorama
     ### managed_logs
```

2. Connect la clé USB à l'appareil.
3. Eteindrel'appliance AWS Panorama.
4. Allumez l'appliance AWS Panorama.
5. L'appareil copie les journaux sur l'appareil. La LED d'étatclignote en bleupendant que cela est en cours.
6. Les fichiers journaux peuvent ensuite être trouvés à l'intérieur`managed_logs`répertoire au format`panorama_device_log_`*`v1_dd_hh_mm`*`.img`

Vous ne pouvez pas déchiffrer l'image du journal vous-même. Travaillez avec le support client, un responsable de compte technique pour AWS Panorama ou un architecte de solutions pour assurer la coordination avec l'équipe de service.

# Surveillance des appliances et des applications avec AmazonCloudWatch

Lorsqu'une appliance est en ligne, AWS Panorama envoie des mesures à AmazonCloudWatch. Vous pouvez créer des graphiques et des tableaux de bord à l'aide de ces mesures dans leCloudWatchpour surveiller l'activité de l'appliance et définir des alarmes qui vous avertissent lorsque les appareils se déconnectent ou que les applications rencontrent des erreurs.

Pour afficher les métriques sur la console CloudWatch

1. Ouverture d'Page Mesures de la console AWS Panorama(`PanoramaDeviceMetrics`namespace).
2. Choisissez un schéma de cotation.
3. Choisissez des métriques pour les ajouter au graphique.
4. Pour choisir une autre statistique et personnaliser le graphique, utilisez les options de l'onglet Graphique des métriques. Par défaut, les graphiques utilisent la statistique `Average` pour toutes les métriques.

> **Tarification**
>
> CloudWatchpossède un niveau Toujours gratuit. Au-delà du seuil de cette offre gratuite, CloudWatch facture les métriques, les tableaux de bord, les alarmes, les journaux et les informations. Pour de plus amples informations, veuillez consulter Tarification CloudWatch.

Pour plus d'informations surCloudWatch, voir laAmazonCloudWatchGuide de l'utilisateur.

Sections

- Utilisation des métriques d'appareil
- Utilisation des mesures d'application
- Configuration des alarmes

## Utilisation des métriques d'appareil

Lorsqu'une appliance est en ligne, elle envoie des métriques à AmazonCloudWatch. Vous pouvez utiliser ces mesures pour surveiller l'activité de l'appareil et déclencher une alarme si les appareils se déconnectent.

- `DeviceActive`— Envoyé périodiquement lorsque l'appareil est actif.

  Dimensions —`DeviceId`et`DeviceName`.

Affichage du`DeviceActive`Métriques avec la fonction`Average`statistique.

## Utilisation des mesures d'application

Lorsqu'une application rencontre une erreur, elle envoie des métriques à AmazonCloudWatch. Vous pouvez utiliser ces mesures pour déclencher une alarme si une application cesse de fonctionner.

- `ApplicationErrors`— Le nombre d'erreurs d'application enregistrées.

  Dimensions —`ApplicationInstanceName`et`ApplicationInstanceId`.

Afficher les métriques d'application grâce à la fonction`Sum`statistique.

## Configuration des alarmes

Pour recevoir des notifications lorsqu'une mesure dépasse un seuil, créez une alarme. Par exemple, vous pouvez créer une alarme qui envoie une notification lorsque la somme des`ApplicationErrors`Métrique reste à 1 pendant 20 minutes.

Pour créer une alarme

1. Ouverture d'AmazonCloudWatchpage Alarmes de console.
2. Choisissez Create alarm (Créer une alarme).
3. ChoisissezSélectionner la métriqueet localisez une mesure pour votre appareil, par exemple`ApplicationErrors`pour`applicationInstance-gk75xmplqbqtenlnmz4ehiu7xa`,`my-application`.
4. Suivez les instructions pour configurer une condition, une action et un nom pour l'alarme.

Pour obtenir les instructions complètes, consultez .Création d'unCloudWatchalarmedans leAmazonCloudWatchGuide de l'utilisateur.

# Résolution des problèmes

Les rubriques suivantes fournissent des conseils de résolution des erreurs et des problèmes que vous pourriez rencontrer lors de l'utilisation de la AWS Panorama console, de l'appliance ou du SDK. Si vous trouvez un problème qui n'est pas répertorié ici, utilisez le bouton Envoyer des commentaires sur cette page pour le signaler.

Vous pouvez trouver les journaux de votre appliance dans la console Amazon CloudWatch Logs. L'appliance télécharge les journaux à partir du code de votre application, du logiciel de l'appliance et des AWS IoT processus au fur et à mesure de leur génération. Pour plus d'informations, veuillez consulter Affichage des journaux AWS Panorama.

## Approvisionnement

Problème : (macOS) Mon ordinateur ne reconnaît pas la clé USB fournie avec un adaptateur USB-C.

Cela peut se produire si vous branchez la clé USB sur un adaptateur USB-C déjà connecté à votre ordinateur. Essayez de déconnecter l'adaptateur et de le reconnecter avec la clé USB déjà connectée.

Problème : le provisionnement échoue lorsque j'utilise ma propre clé USB.

Problème : le provisionnement échoue lorsque j'utilise le port USB 2.0 de l'appliance.

L'AWS Panoramaappliance est compatible avec les dispositifs de mémoire flash USB d'une capacité comprise entre 1 et 32 Go, mais tous ne sont pas compatibles. Certains problèmes ont été observés lors de l'utilisation du port USB 2.0 pour le provisionnement. Pour des résultats homogènes, utilisez la clé USB incluse avec le port USB 3.0 (à côté du port HDMI).

Pour le Lenovo ThinkEdge® SE70, aucune clé USB n'est fournie avec l'appareil. Utilisez une clé USB 3.0 avec au moins 1 Go de stockage.

## Configuration de l'appliance

Problème : l'appliance affiche un écran vide lors du démarrage.

Après avoir terminé la séquence de démarrage initiale, qui prend environ une minute, l'appliance affiche un écran vide pendant une minute ou plus pendant qu'elle charge votre modèle et démarre

votre application. En outre, l'appliance ne produit pas de vidéo si vous connectez un écran après son allumage.

Problème : L'appareil ne répond pas lorsque je maintiens le bouton d'alimentation enfoncé pour l'éteindre.

L'appareil met jusqu'à 10 secondes pour s'éteindre en toute sécurité. Vous devez maintenir le bouton d'alimentation enfoncé pendant une seconde seulement pour démarrer la séquence d'arrêt. Pour une liste complète des opérations sur les boutons, voirBoutons et voyants de l'appliance AWS Panorama.

Problème : je dois générer une nouvelle archive de configuration pour modifier les paramètres ou remplacer un certificat perdu.

AWS Panoramane stocke pas le certificat de l'appareil ou la configuration réseau une fois que vous l'avez téléchargé, et vous ne pouvez pas réutiliser les archives de configuration. Supprimez l'appliance à l'aide de la AWS Panorama console et créez-en une nouvelle avec une nouvelle archive de configuration.

## Configuration de l'application

Problème : Lorsque j'exécute plusieurs applications, je ne peux pas contrôler celle qui utilise la sortie HDMI.

Lorsque vous déployez plusieurs applications dotées de nœuds de sortie, l'application qui a démarré le plus récemment utilise la sortie HDMI. Si cette application cesse de fonctionner, une autre application peut utiliser le résultat. Pour n'autoriser qu'une seule application à accéder à la sortie, supprimez le nœud de sortie et le bord correspondant du manifeste d'application de l'autre application, puis redéployez-les.

Problème : le résultat de l'application n'apparaît pas dans les journaux

Configurez un enregistreur Python dans lequel écrire des fichiers journaux. `/opt/aws/panorama/logs` Ils sont capturés dans un flux de journal pour le nœud du conteneur de code. Les flux de sortie et d'erreur standard sont capturés dans un flux de journal distinct appelé`console-output`. Si vous l'utilisez`print`, utilisez l'`flush=True`option pour empêcher les messages de rester bloqués dans la mémoire tampon de sortie.

Erreur : You've reached the maximum number of versions for package SAMPLE_CODE. Deregister unused package versions and try again.

Source : AWS Panorama service

Chaque fois que vous déployez une modification dans une application, vous enregistrez une version du correctif qui représente la configuration du package et les fichiers de ressources pour chaque package utilisé. Utilisez le script de correctifs de nettoyage pour désenregistrer les versions de correctifs non utilisées.

## Streams de caméras

Erreur : liveMedia0: Failed to get SDP description: Connection to server failed: Connection timed out (-115)

Erreur : liveMedia0: Failed to get SDP description: 404 Not Found; with the result code: 404

Erreur : liveMedia0: Failed to get SDP description: DESCRIBE send() failed: Broken pipe; with the result code: -32

Source : journal des nœuds de caméra

L'appliance ne peut pas se connecter au flux de caméra de l'application. Dans ce cas, la sortie vidéo est vide ou se fige sur la dernière image traitée pendant que l'application attend une image vidéo provenant du SDK de l'AWS Panoramaapplication. Le logiciel de l'appliance tente de se connecter au flux de caméra et enregistre les erreurs de temporisation dans le journal du nœud de caméra. Vérifiez que l'URL du flux de votre caméra est correcte et que le trafic RTSP est routable entre la caméra et l'appliance au sein de votre réseau. Pour plus d'informations, veuillez consulter Connexion de l'appliance AWS Panorama à votre réseau.

Erreur : ERROR finalizeInterface(35) Camera credential fetching for port [username] failed

Source : journal de l'OCC

Le AWS Secrets Manager secret des informations d'identification du flux de caméra est introuvable. Supprimez le flux de caméra et recréez-le.

Erreur : Camera did not provide an H264 encoded stream

Source : journal des nœuds de caméra

Le flux de caméra possède un encodage autre que H.264, tel que H.265. Redéployez l'application avec un flux de caméra H.264. Pour plus de détails sur les appareils photo pris en charge, consultezCaméras compatibles.

# Services de sécurité AWS Panorama

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. En tant que client AWS, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entre AWS et vous-même. Le modèle de responsabilité partagée décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- Sécurité du cloud – AWS est responsable de la protection de l'infrastructure qui exécute des services AWS dans le cloud AWS. AWS vous fournit également les services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de conformité AWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Panorama, consultezAWSServices concernés par le programme de conformité.
- Sécurité dans le cloud : votre responsabilité est déterminée par le service AWS que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lorsque vous utilisez AWS Panorama. Les rubriques suivantes montrent comment configurer AWS Panorama pour répondre à vos objectifs de sécurité et de conformité. Vous pouvez également apprendre à utiliser d'autres services AWS qui vous permettent de surveiller et de sécuriser vos ressources AWS Panorama.

Rubriques

# Caractéristiques de sécurité AWS Panorama Appliance

Pour protéger votreapplications, modèles, et du matériel contre le code malveillant et d'autres exploits, l'appliance AWS Panorama met en œuvre un ensemble complet de fonctionnalités de sécurité. Notalement (entre autres) :

- Chiffrement de disque complet— La solution matérielle-logicielle implémente le chiffrement complet du disque complet LUKS2 (Linux Unified Key Setup). Toutes les données du logiciel système et des applications sont chiffrées à l'aide d'une clé spécifique à votre appareil. Même avec un accès physique à l'appareil, un attaquant ne peut pas inspecter le contenu de son stockage.
- Disposition de la mémoire de façon aléatoire— Pour se protéger contre les attaques qui ciblent le code exécutable chargé en mémoire, AWS Panorama Appliance utilise la randomisation de la disposition de l'espace d'adressage (ASLR). ASLR réalise aléatoirement l'emplacement du code du système d'exploitation lorsqu'il est chargé en mémoire. Cela empêche l'utilisation d'exploits qui tentent d'écraser ou d'exécuter des sections spécifiques de code en prédisant où il est stocké au moment de l'exécution.
- environnement d'exécution de confiance— L'appliance utilise un environnement d'exécution approuvé (TEE) basé sur ARM TrustZone, avec des ressources de stockage, de mémoire et de traitement isolées. Les clés et autres données sensibles stockées dans la zone de confiance ne sont accessibles que par une application de confiance, qui s'exécute dans un système d'exploitation distinct au sein du TEE. Le logiciel AWS Panorama Appliance s'exécute dans un environnement Linux non fiable en même temps que le code d'application. Il ne peut accéder aux opérations cryptographiques qu'en envoyant une demande à l'application sécurisée.
- provisionnement sécurisé— Lorsque vous provisionnez une appliance, les informations d'identification (clés, certificats et autres éléments cryptographiques) que vous transférez vers l'appareil ne sont valides que pendant une courte période. L'appliance utilise les informations d'identification de courte durée pour se connecter àAWS IoTet demande pour lui-même un certificat valide plus longtemps. Le service AWS Panorama génère des informations d'identification et les chiffre à l'aide d'une clé codée en dur sur l'appareil. Seul l'appareil qui a demandé le certificat peut le déchiffrer et communiquer avec AWS Panorama.
- Démarrage sécurisé— Lorsque l'appareil démarre, chaque composant logiciel est authentifié avant son exécution. La ROM de démarrage, logiciel codé en dur dans le processeur qui ne peut pas être modifié, utilise une clé de chiffrement codée en dur pour déchiffrer le chargeur de démarrage, qui valide le noyau de l'environnement d'exécution approuvé, etc.

- Noyau signé— Les modules du noyau sont signés avec une clé de chiffrement asymétrique. Le noyau du système d'exploitation déchiffre la signature avec la clé publique et vérifie qu'elle correspond à la signature du module avant de charger le module en mémoire.
- dm-verity— Comme pour la validation des modules du noyau, la solution matérielle-logicielle utilise le module Linux Device Mapper`dm-verity`pour vérifier l'intégrité de l'image du logiciel de l'appliance avant de la monter. Si le logiciel de l'appliance est modifié, il ne sera pas exécuté.
- Prévention du restauration— Lorsque vous mettez à jour le logiciel de l'appliance, l'appliance souffle un fusible électronique sur le SoC (système sur une puce). Chaque version du logiciel s'attend à ce qu'un nombre croissant de fusibles soit explosé et ne peut pas fonctionner si d'autres sont soufflés.

# Bonnes pratiques de sécurité AWS Panorama Appliance

Gardez à l'esprit les meilleures pratiques suivantes lorsque vous utilisez l'appliance AWS Panorama.

- Sécurisez physiquement l'appliance— Installez l'appliance dans un rack de serveur fermé ou une pièce sécurisée. Limitez l'accès physique à l'appareil au personnel autorisé.
- Sécurisez la connexion réseau de l'appliance: Connect l'appliance à un routeur qui limite l'accès aux ressources internes et externes. L'appliance doit se connecter à des caméras, qui peuvent se trouver sur un réseau interne sécurisé. Il doit également se connecter àAWS. Utilisez le deuxième port Ethernet uniquement pour la redondance physique et configurez le routeur pour autoriser uniquement le trafic requis.

  Utilisez l'une des configurations réseau recommandées pour planifier la disposition de votre réseau. Pour plus d'informations, consultez Connexion de l'appliance AWS Panorama à votre réseau.
- Formater la clé USB— Après avoir provisionné une appliance, retirez la clé USB et formatez-la. L'appliance n'utilise pas la clé USB après s'être enregistrée auprès du service AWS Panorama. Formatez le lecteur pour supprimer les informations d'identification temporaires, les fichiers de configuration et les journaux de provisionnement.
- Garder l'appareil à jour— Appliquez les mises à jour logicielles du dispositif en temps opportun. Lorsque vous affichez une appliance dans la console AWS Panorama, la console vous avertit si une mise à jour logicielle est disponible. Pour plus d'informations, consultez Gestion d'une appliance AWS Panorama.

  Avec l'DescribeDeviceFonctionnement de l'API, vous pouvez automatiser la vérification des mises à jour en comparant le`LatestSoftware`et`CurrentSoftware`. Lorsque la dernière version du logiciel diffère de la version actuelle, appliquez la mise à jour avec la console ou en utilisant leCréer un travail pour les appareils.
- Si vous arrêtez d'utiliser une appliance, réinitialisez-la— Avant de déplacer l'appliance hors de votre centre de données sécurisé, réinitialisez-la complètement. Lorsque l'appareil est hors tension et branché, appuyez simultanément sur le bouton d'alimentation et de réinitialisation pendant 5 secondes. Cela supprime les informations d'identification du compte, les applications et les journaux de la solution matérielle-logicielle.

  Pour plus d'informations, consultez Boutons et voyants de l'appliance AWS Panorama.

- Limiter l'accès à AWS Panorama et à d'autres services AWS— LeAWSPanoramaFullAccessfournit un accès à toutes les opérations de l'API AWS Panorama et, le cas échéant, à d'autres services. Dans la mesure du possible, la stratégie limite l'accès aux ressources en fonction des conventions de dénomination. Par exemple, il permet d'accéder àAWS Secrets Managersecrets dont les noms commencent par`panorama`. Pour les utilisateurs qui ont besoin d'un accès en lecture seule ou d'un accès à un ensemble de ressources plus spécifique, utilisez la stratégie gérée comme point de départ de vos stratégies de moins-privilèges.

  Pour plus d'informations, consultez Politiques IAM basées sur l'identité pour AWS Panorama.

# Protection des données dans AWS Panorama

Le modèle de responsabilité AWS partagée s'applique à la protection des données dans AWS Panorama. Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure globale sur laquelle l'ensemble du AWS Cloud s'exécute. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour en savoir plus sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWSet RGPD (Règlement général sur la protection des données) sur le AWSBlog de sécurité.

À des fins de protection des données, nous vous recommandons de protéger les informations d'identification Compte AWS et de configurer les comptes utilisateur individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez les certificats SSL/TLS pour communiquer avec les ressources AWS. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez une API (Interface de programmation) et le journal de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de chiffrement AWS, ainsi que tous les contrôles de sécurité par défaut au sein des Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés FIPS (Federal Information Processing Standard) 140-2 lorsque vous accédez à AWS via une CLI (Interface de ligne de commande) ou une API (Interface de programmation), utilisez un point de terminaison FIPS (Federal Information Processing Standard). Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que

le champ Name (Nom). Cela inclut lorsque vous travaillez avec AWS Panorama ou une autre solution Services AWS à l'aide de la console, de l'API ou AWS des kits SDK. AWS CLI Toutes les données que vous saisissez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Sections



## Chiffrement en transit

Les points de terminaison de l'API AWS Panorama prennent en charge les connexions sécurisées uniquement via HTTPS. Lorsque vous gérez les ressources AWS Panorama à l'aide du AWS Management Console SDK AWS ou de l'API AWS Panorama, toutes les communications sont cryptées avec le protocole TLS (Transport Layer Security). Les communications entre l'appliance AWS Panorama et AWS sont également cryptées avec le protocole TLS. Les communications entre l'appliance AWS Panorama et les caméras via RTSP ne sont pas cryptées.

Pour obtenir la liste complète des points de terminaison d'API, consultez la section Régions et points de terminaison AWS dans le. Références générales AWS

## Appliance AWS Panorama

L'appliance AWS Panorama possède des ports physiques pour l'Ethernet, la vidéo HDMI et le stockage USB. Le lecteur de carte SD, le Wi-Fi et le Bluetooth ne sont pas utilisables. Le port USB est uniquement utilisé lors du provisionnement pour transférer une archive de configuration vers l'appliance.

Le contenu de l'archive de configuration, qui inclut le certificat de provisionnement et la configuration réseau de l'appliance, n'est pas chiffré. AWS Panorama ne stocke pas ces fichiers ; ils ne peuvent être récupérés que lorsque vous enregistrez une appliance. Après avoir transféré l'archive de

configuration vers une appliance, supprimez-la de votre ordinateur et de votre périphérique de stockage USB.

L'ensemble du système de fichiers de l'appliance est crypté. En outre, l'appliance applique plusieurs protections au niveau du système, notamment la protection anti-annulation pour les mises à jour logicielles requises, le noyau signé et le chargeur de démarrage, ainsi que la vérification de l'intégrité du logiciel.

Lorsque vous arrêtez d'utiliser l'appliance, effectuez une réinitialisation complète pour supprimer les données de l'application et réinitialiser le logiciel de l'appliance.

## Applications

Vous contrôlez le code que vous déployez sur votre appliance. Validez tout le code de l'application pour détecter les problèmes de sécurité avant de le déployer, quelle que soit sa source. Si vous utilisez des bibliothèques tierces dans votre application, examinez attentivement les politiques de licence et de support de ces bibliothèques.

L'utilisation du processeur, de la mémoire et du disque de l'application n'est pas limitée par le logiciel de l'appliance. Une application utilisant trop de ressources peut avoir un impact négatif sur les autres applications et sur le fonctionnement de l'appareil. Testez les applications séparément avant de les combiner ou de les déployer dans des environnements de production.

Les ressources de l'application (codes et modèles) ne sont pas isolées de l'accès au sein de votre compte, de votre appliance ou de votre environnement de construction. Les images de conteneur et les archives de modèles générées par la CLI de l'application AWS Panorama ne sont pas chiffrées. Utilisez des comptes distincts pour les charges de travail de production et n'autorisez l'accès qu'en cas de besoin.

## Autres services

Pour stocker vos modèles et vos conteneurs d'applications en toute sécurité dans Amazon S3, AWS Panorama utilise le chiffrement côté serveur avec une clé gérée par Amazon S3. Pour plus d'informations, consultez la section Protection des données à l'aide du chiffrement dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Les informations d'identification du flux de caméra sont cryptées au reposAWS Secrets Manager. Le rôle IAM de l'appliance lui donne l'autorisation de récupérer le secret afin d'accéder au nom d'utilisateur et au mot de passe du flux.

L'appliance AWS Panorama envoie les données des journaux à Amazon CloudWatch Logs. CloudWatch Les journaux chiffrent ces données par défaut et peuvent être configurés pour utiliser une clé gérée par le client. Pour plus d'informations, consultez la section Chiffrer les données des CloudWatch journaux dans les journaux à l'aide AWS KMS du guide de l'utilisateur Amazon CloudWatch Logs.

# Gestion des identités et des accès pour AWS Panorama

AWS Identity and Access Management (IAM) est un Service AWS qui aide un administrateur à contrôler en toute sécurité l'accès aux ressources AWS. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources AWS Panorama. IAM est un Service AWS que vous pouvez utiliser sans frais supplémentaires.

Rubriques



## Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans AWS Panorama.

Utilisateur du service : si vous utilisez le service AWS Panorama pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités d'AWS Panorama pour effectuer votre travail, il se peut que vous ayez besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité d'AWS Panorama, consultezRésolution des problèmes d'identité et d'accès à AWS Panorama.

Administrateur du service — Si vous êtes responsable des ressources AWS Panorama au sein de votre entreprise, vous avez probablement un accès complet à AWS Panorama. Il vous incombe de déterminer les fonctionnalités et ressources d'AWS Panorama auxquelles les utilisateurs de votre

service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec AWS Panorama, consultezComment AWS Panorama fonctionne avec IAM.

Administrateur IAM : si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à AWS Panorama. Pour consulter des exemples de politiques basées sur l'identité AWS Panorama que vous pouvez utiliser dans IAM, consultez. Exemples de politiques basées sur l'identité AWS Panorama

## Authentification par des identités

L'authentification correspond au processus par lequel vous vous connectez à AWS avec vos informations d'identification. Vous devez vous authentifier (être connecté à AWS) en tant qu'Utilisateur racine d'un compte AWS, en tant qu'utilisateur IAM ou en endossant un rôle IAM.

Vous pouvez vous connecter à AWS en tant qu'identité fédérée à l'aide des informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification de connexion unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS en utilisant la fédération, vous endossez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter à la AWS Management Console ou au portail d'accès AWS. Pour plus d'informations sur la connexion à AWS, consultez Connexion à votre Compte AWSdans le Guide de l'utilisateurConnexion à AWS.

Si vous accédez à AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes en utilisant vos informations d'identification. Si vous n'utilisez pas les outils AWS, vous devez signer les requêtes vous-même. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer des demandes vous-même, consultez Signature des demandes d'API AWS dans le Guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, AWS vous recommande d'utiliser l'authentification multifactorielle (MFA) pour améliorer la sécurité de votre compte. Pour en savoir

plus, veuillez consulter Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Utilisation de l'authentification multifactorielle (MFA) dans l'interface AWS dans le Guide de l'utilisateur IAM.

### Utilisateur root Compte AWS

Lorsque vous créez un Compte AWS, vous commencez avec une seule identité de connexion disposant d'un accès complet à tous les Services AWS et ressources du compte. Cette identité est appelée utilisateur root du Compte AWS. Vous pouvez y accéder en vous connectant à l'aide de l'adresse électronique et du mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur root pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur root et utilisez-les pour effectuer les tâches que seul l'utilisateur root peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez Tâches nécessitant les informations d'identification de l'utilisateur root dans le Guide de l'utilisateur IAM.

### Utilisateurs et groupes IAM

Un utilisateur IAM est une identité dans votre Compte AWS qui dispose d'autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

## Rôles IAM

Un rôle IAM est une entité au sein de votre Compte AWS qui dispose d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez temporairement endosser un rôle IAM dans la AWS Management Console en changeant de rôle. Vous pouvez obtenir un rôle en appelant une opération d'API AWS CLI ou AWS à l'aide d'une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Utilisation de rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center.
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, certains Services AWS vous permettent d'attacher une politique directement à une ressource (au lieu d'utiliser un rôle en tant que proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.
- Accès interservices : certains Services AWS utilisent des fonctions dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, une fonction du service ou un rôle lié au service.
  - Forward access sessions (FAS) – Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions dans AWS, vous êtes considéré comme un principal. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui déclenche une autre action dans un

autre service. FAS utilise les autorisations du principal appelant Service AWS, combinées à la demande Service AWS pour effectuer des demandes aux services en aval. Les demandes de FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez Sessions de transmission d'accès.

- Fonction du service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
- Rôle lié au service – Un rôle lié au service est un type de fonction du service lié à un Service AWS. Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service s'affichent dans votre Compte AWS et sont détenus par le service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications s'exécutant sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer des informations d'identification temporaires pour les applications s'exécutant sur une instance EC2 et effectuant des demandes d'API AWS CLI ou AWS. Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un rôle AWS à une instance EC2 et le rendre disponible à toutes les applications associées, vous pouvez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez Quand créer un rôle IAM (au lieu d'un utilisateur) dans le Guide de l'utilisateur IAM.

## Gestion des accès à l'aide de politiques

Vous contrôlez les accès dans AWS en créant des politiques et en les attachant à des identités AWS ou à des ressources. Une politique est un objet dans AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit les autorisations de ces dernières. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur racine ou séance de rôle) envoie une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques

sont stockées dans AWS en tant que documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Présentation des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques JSON AWS pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur avec cette politique peut obtenir des informations utilisateur à partir de la AWS Management Console, de la AWS CLI ou de l'API AWS.

### Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Création de politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez attacher à plusieurs utilisateurs, groupes et rôles dans votre Compte AWS. Les politiques gérées incluent les politiques gérées par AWS et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

### politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment Amazon S3. Dans les services qui sont

compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou des Services AWS.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques gérées AWS depuis IAM dans une politique basée sur une ressource.

### Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3, AWS WAF et Amazon VPC sont des exemples de services prenant en charge les ACL. Pour en savoir plus sur les listes de contrôle d'accès, consultez Présentation des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

### Autres types de politique

AWS prend en charge d'autres types de politiques moins courantes. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d'autorisations : une limite d'autorisations est une fonction avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations qui en résultent représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCP) - les SCP sont des politiques JSON qui spécifient le nombre maximal d'autorisations pour une organisation ou une unité d'organisation (OU) dans AWS

Organizations. AWS Organizations est un service qui vous permet de regrouper et de gérer de façon centralisée plusieurs Comptes AWS détenus par votre entreprise. Si vous activez toutes les fonctions d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. La SCP limite les autorisations pour les entités dans les comptes membres, y compris dans chaque Utilisateur racine d'un compte AWS. Pour plus d'informations sur les organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations.

- politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de la séance obtenue sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations, consultez Politiques de séance dans le Guide de l'utilisateur IAM.

### Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations obtenues sont plus compliquées à comprendre. Pour découvrir la façon dont AWS détermine s'il convient d'autoriser une demande en présence de plusieurs types de politiques, veuillez consulter Logique d'évaluation de politiques dans le Guide de l'utilisateur IAM.

## Comment AWS Panorama fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS Panorama, vous devez connaître les fonctionnalités IAM disponibles avec AWS Panorama. Pour obtenir une vue d'ensemble de la manière dont AWS Panorama et les autres AWS services fonctionnent avec IAM, consultez les AWSservices compatibles avec IAM dans le guide de l'utilisateur d'IAM.

Pour un aperçu des autorisations, des politiques et des rôles tels qu'ils sont utilisés par AWS Panorama, consultezAutorisations AWS Panorama.

## Exemples de politiques basées sur l'identité AWS Panorama

Par défaut, les utilisateurs et les rôles IAM ne sont pas autorisés à créer ou à modifier les ressources AWS Panorama. Ils ne peuvent pas non plus exécuter des tâches à l'aide de AWS Management Console, AWS CLI ou de l'API AWS. Un administrateur IAM doit créer des politiques IAM autorisant

les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces politiques aux utilisateurs ou aux groupes IAM ayant besoin de ces autorisations.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, veuillez consulter Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console AWS Panorama
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

## Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources AWS Panorama dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l’identité, suivez ces instructions et recommandations :

- Démarrer avec AWS gérées et évoluez vers les autorisations de moindre privilège - Pour commencer à accorder des autorisations à vos utilisateurs et charges de travail, utilisez les politiques gérées AWS qui accordent des autorisations dans de nombreux cas d’utilisation courants. Elles sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client AWS qui sont spécifiques à vos cas d’utilisation. Pour de plus amples informations, consultez AWS Politiques gérées ou AWS Politiques gérées pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège - Lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l’exécution d’une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d’informations sur l’utilisation d’IAM pour appliquer des autorisations, consultez Politiques et autorisations dans IAM dans le Guide de l’utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l’accès - Vous pouvez ajouter une condition à vos politiques afin de limiter l’accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder

l'accès aux actions de service si elles sont utilisées via un Service AWS spécifique, comme AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles - IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Authentification multifactorielle (MFA) nécessaire : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root dans votre Compte AWS, activez l'authentification multifactorielle pour une sécurité renforcée. Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Configuration de l'accès aux API protégé par MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

## Utilisation de la console AWS Panorama

Pour accéder à la console AWS Panorama, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources AWS Panorama de votre AWS compte. Si vous créez une politique basée sur l'identité qui est plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs et rôles IAM) tributaires de cette politique.

Pour de plus amples informations, veuillez consulter la page Politiques IAM basées sur l'identité pour AWS Panorama.

## Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations nécessaires pour réaliser cette action sur la console ou par programmation à l'aide de l'AWS CLI ou de l'API AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

## AWSpolitiques gérées pour AWS Panorama

UnAWSune politique gérée est une politique autonome créée et administrée parAWS.AWSles politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

Gardez à l'esprit queAWSles politiques gérées peuvent ne pas accorder les autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont disponibles pour tousAWSclients à

utiliser. Nous vous recommandons de réduire davantage les autorisations en définissantpolitiques gérées par le clientqui sont spécifiques à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les stratégies gérées par AWS. SiAWSmet à jour les autorisations définies dans unAWSpolitique gérée, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée.AWSest le plus susceptible de mettre à jour unAWSpolitique gérée lors d'une nouvelleService AWSest lancée ou de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez la rubrique Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

AWS Panorama fournit les politiques gérées suivantes. Pour le contenu complet et l'historique des modifications de chaque politique, consultez les pages liées dans la console IAM.

- AWSPanoramaFullAccess— Fournit un accès complet à AWS Panorama, aux points d'accès AWS Panorama dans Amazon S3, aux informations d'identification des appliances dansAWS Secrets Manager, et les journaux des appareils sur AmazonCloudWatch. Inclut l'autorisation de créer unrôle lié à un servicepour AWS Panorama.
- AWSPanoramaServiceLinkedRolePolicy— Permet à AWS Panorama de gérer les ressources dans AWS IoT, AWS Secrets Manager et AWS Panorama.
- AWSPanoramaApplianceServiceRolePolicy— Permet à une appliance AWS Panorama de télécharger des journaux versCloudWatch, et pour obtenir des objets à partir des points d'accès Amazon S3 créés par AWS Panorama.

## Mises à jour d'AWS Panorama pourAWSpolitiques gérées

Le tableau suivant décrit les mises à jour des politiques gérées pour AWS Panorama.

| Modification | Description | Date |
| --- | --- | --- |
| AWSPanoramaFullAccess - mise à jour d'une politique existante | Des autorisations ont été ajoutées à la politique utilisateur pour permettre aux utilisateurs de consulter les groupes de journaux dans leCloudWatchConsole de journalisation. | 13/01/2022 |

| Modification | Description | Date |
| --- | --- | --- |
| AWSPanoramaFullAccess : mise à jour d'une politique existante | Ajout d'autorisations à la politique utilisateur pour permettre aux utilisateurs de gérer l'AWS Panoramarôle lié à un service, et pour accéder aux ressources AWS Panorama dans d'autres services, notamment IAM, Amazon S3,CloudWatch, et Secrets Manager. | 20/10/2021 |
| AWSPanoramaApplianceServiceRolePolicy – Nouvelle politique | Nouvelle politique pour le rôle de service d'AWS Panorama Appliance | 20/10/2021 |
| AWSPanoramaServiceLinkedRolePolicy – Nouvelle politique | Nouvelle politique pour le rôle lié au service AWS Panorama. | 20/10/2021 |
| AWS Panorama a commencé à suivre les modifications | AWS Panorama a commencé à suivre les modifications apportées à sonAWSpolitiques gérées. | 20/10/2021 |

## Utilisation des rôles liés à un service pour AWS Panorama

AWS Panorama utilise des rôles AWS Identity and Access Management (IAM) liés à un service. Un rôle lié à un service est un type unique de rôle IAM lié directement à AWS Panorama. Les rôles liés à un service sont prédéfinis par AWS Panorama et comprennent toutes les autorisations nécessaires au service pour appeler d'autres services AWS en votre nom.

Un rôle lié à un service permet d'utiliser AWS Panorama plus facilement, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. AWS Panorama définit les autorisations de ses rôles liés à un service et, sauf définition contraire, seul AWS Panorama peut endosser ses rôles. Les

autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Vos ressources AWS Panorama sont ainsi protégées, car vous ne pouvez pas involontairement supprimer l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM et recherchez les services avec un Oui dans la colonne Rôle lié à un service. Choisissez un Oui ayant un lien permettant de consulter la documentation du rôle lié à un service, pour ce service.

Sections



## Autorisations des rôles liés à un service pour AWS Panorama

AWS Panoramautilise le rôle lié à un service nommé .RÃ´le de service AWS pour AWS Panorama—Permet à AWS Panorama de gérer les ressources dans AWS IoT, AWS Secrets Manager et AWS Panorama.

Le rôle lié à un service AWSServiceRoleForAWSServiceRoleForAWSSorama approuvent les services suivants pour assumer le rôle :

- `panorama.amazonaws.com`

La stratégie d'autorisations liée au rôle permet à AWS Panorama de réaliser les actions suivantes :

- ContrôleAWS Panoramaressources
- GérerAWS IoTressources pour leAWS PanoramaAppliance
- Accès àAWS Secrets Managersecrets pour obtenir les informations d'identification de caméra

Pour obtenir la liste complète des autorisations,afficher la stratégie de rôle de connexion AWS Panorama Servicedans la console IAM.

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, un groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM.

## Création d'un rôle lié à un service pour AWS Panorama

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous enregistrez une appliance dans leAWS Management Console, leAWS CLI, ou leAWSAPI,AWS Panoramacrée automatiquement le rôle lié au service.

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous enregistrez une appliance,AWS Panoramacrée à nouveau le rôle lié au service à votre place.

## Modification d'un rôle lié à un service pour AWS Panorama

AWS Panoramane vous permet pas de modifier le rôle lié à un service AWSServiceRoleForAWSSorama. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas modifier le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour en savoir plus, consultez Modification d'un rôle lié à un service dans le guide de l'utilisateur IAM.

## Suppression d'un rôle lié à un service pour AWS Panorama

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

Pour supprimer leAWS Panoramales ressources utilisées par AWSServiceRoleForAWSSorama utilisent les procédures décrites dans les sections suivantes de ce guide.

- Supprimer des versions et des applications
- Annuler l'enregistrement d'un appareil

> **Note**
>
> Si le service AWS Panorama utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer le rôle lié à un service AWSServiceRoleForAWSPanorama, utilisez la console IAM, leAWS CLI, ou leAWSAPI. Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

### Régions prises en charge pour les rôles liés à un service AWS Panorama

AWS Panorama prend en charge l'utilisation des rôles liés à un service dans toutes les régions où le service est disponible. Pour de plus amples informations, veuillez consulterAWSRégions et points de terminaison.

## Prévention du député confus entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. Dans AWS, l'emprunt d'identité entre services peut entraîner le problème de député confus. L'emprunt d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé pour utiliser ses autorisations afin d'agir sur les ressources d'un autre client de manière à ce qu'il ne soit pas autorisé à y accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous vous recommandons d'utiliser les clés de contexte de condition globale `aws:SourceArn` et `aws:SourceAccount` dans les politiques de ressources pour limiter les autorisations à la ressource octroyées par AWS Panorama à un autre service. Si vous utilisez les deux clés de contexte de condition globale, la valeur `aws:SourceAccount` et le compte de la valeur `aws:SourceArn` doit utiliser le même ID de compte lorsqu'il est utilisé dans la même déclaration de politique.

Pour`aws:SourceArn`doit être l'ARN d'unAWS Panoramaappareil.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez

la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:servicename::123456789012:*`.

Pour obtenir des instructions sur la sécurisation du rôle de serviceAWS Panoramapermet de donner la permission à laAWS PanoramaAppliance, voirSécurisation du rôle d'appliance.

## Résolution des problèmes d'identité et d'accès à AWS Panorama

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS Panorama et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans AWS Panorama
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources AWS Panorama

### Je ne suis pas autorisé à effectuer une action dans AWS Panorama

Si la AWS Management Console indique que vous n'êtes pas autorisé à exécuter une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni votre nom d'utilisateur et votre mot de passe.

L'exemple d'erreur suivant se produit lorsque l'utilisateur `mateojackson` IAM essaie d'utiliser la console pour consulter les détails d'une appliance mais ne dispose pas des `panorama:DescribeAppliance` autorisations nécessaires.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 panorama:DescribeAppliance on resource: my-appliance
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la ressource `my-appliance` à l'aide de l'action `panorama:DescribeAppliance`.

### Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à AWS Panorama.

Certains Services AWS vous permettent de transmettre un rôle existant à ce service, au lieu de créer une nouvelle fonction du service ou rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans AWS Panorama. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction du service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez encore besoin d'aide, contactez votre administrateur AWS. Votre administrateur vous a fourni vos informations de connexion.

## Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources AWS Panorama

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si AWS Panorama prend en charge ces fonctionnalités, consultezComment AWS Panorama fonctionne avec IAM.
- Pour savoir comment octroyer l'accès à vos ressources à des Comptes AWS dont vous êtes propriétaire, consultez la section Fournir l'accès à un utilisateur IAM dans un autre Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment octroyer l'accès à vos ressources à des tiers Comptes AWS, consultez Fournir l'accès aux Comptes AWS appartenant à des tiers dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.

- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

# Validation de conformité pour AWS Panorama

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides de démarrage rapide sur la sécurité et la conformité : ces guides de déploiement abordent les considérations architecturales et fournissent des étapes pour déployer des environnements de base axés sur AWS la sécurité et la conformité.
- Architecture axée sur la sécurité et la conformité HIPAA sur Amazon Web Services : ce livre blanc décrit comment les entreprises peuvent créer des applications AWS conformes à la loi HIPAA.

  > **Note**
  >
  > Tous ne Services AWS sont pas éligibles à la loi HIPAA. Pour plus d'informations, consultez le HIPAA Eligible Services Reference.

- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Guides de conformité destinés aux clients — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans plusieurs cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).
- Évaluation des ressources à l'aide des règles du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.

- AWS Security Hub— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez Référence des contrôles Security Hub.
- Amazon GuardDuty — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.
- AWS Audit Manager— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

## Considérations supplémentaires concernant la présence de personnes

Vous trouverez ci-dessous quelques bonnes pratiques à prendre en compte lors de l'utilisation d'AWS Panorama dans le cadre de scénarios impliquant la présence éventuelle de personnes :

- Assurez-vous de connaître et de respecter toutes les lois et réglementations applicables à votre cas d'utilisation. Cela peut inclure les lois relatives au positionnement et au champ de vision de vos caméras, les exigences en matière de notification et de signalisation lors du placement et de l'utilisation des caméras, ainsi que les droits des personnes susceptibles d'être présentes dans vos vidéos, y compris leur droit à la vie privée.
- Tenez compte de l'effet de vos caméras sur les personnes et leur vie privée. Outre les exigences légales, déterminez s'il serait approprié d'apposer un avis dans les zones où se trouvent vos caméras, et si les caméras doivent être placées bien en vue et exemptes de toute occlusion, afin que les gens ne soient pas surpris qu'ils soient devant la caméra.
- Mettez en place des politiques et des procédures appropriées pour le fonctionnement de vos caméras et pour l'examen des données obtenues par les caméras.
- Tenez compte des contrôles d'accès, des limites d'utilisation et des périodes de conservation appropriés pour les données obtenues à partir de vos caméras.

# Sécurité de l'infrastructure dans AWS Panorama

En tant que service géré, AWS Panorama est protégé parAWSsécurité du réseau mondial. Pour plus d'informations sur les services de sécurité AWS et la manière dont AWS protège l'infrastructure, consultez la section Sécurité du cloud AWS. Pour concevoir votre environnement AWS en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le Security Pillar AWS Well-Architected Framework (Pilier de sécurité de l'infrastructure Well-Architected Framework).

Vous utilisezAWSappels d'API publiés pour accéder à AWS Panorama via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et nous recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

## Déploiement de l'appliance AWS Panorama dans votre centre de données

L'appliance AWS Panorama a besoin d'un accès Internet pour communiquer avecAWSservices. Il doit également avoir accès à votre réseau interne de caméras. Il est important de bien réfléchir à la configuration de votre réseau et de ne fournir à chaque appareil que l'accès dont il a besoin. Faites attention si votre configuration permet à l'appliance AWS Panorama de servir de pont vers un réseau de caméras IP sensible.

Vous êtes responsable de ce qui suit :

- La sécurité réseau physique et logique de l'appliance AWS Panorama.
- Utilisation sécurisée des caméras connectées au réseau lorsque vous utilisez l'appliance AWS Panorama.
- Maintien à jour de l'appliance AWS Panorama et du logiciel de caméra.

- Respecter toutes les lois ou réglementations applicables associées au contenu des vidéos et des images que vous collectez dans vos environnements de production, y compris celles relatives à la confidentialité.

L'appliance AWS Panorama utilise des flux de caméra RTSP non chiffrés. Pour plus d'informations sur la connexion de l'appliance AWS Panorama à votre réseau, consultezConnexion de l'appliance AWS Panorama à votre réseau. Pour plus de détails sur le chiffrement, voirProtection des données dans AWS Panorama.

# Logiciel d'environnement d'exécution dans AWS Panorama

AWS Panorama fournit un logiciel qui exécute le code de votre application dans un environnement basé sur Ubuntu Linux sur AWS Panorama Appliance. AWS Panorama est responsable de la mise à jour des logiciels de l'image de l'appliance. AWS Panorama publie régulièrement des mises à jour logicielles, que vous pouvez appliquer enutilisation de la console AWS Panorama.

Vous pouvez utiliser des bibliothèques dans le code de votre application en les installant dans le`Dockerfile`. Pour garantir la stabilité des applications dans toutes les versions, choisissez une version spécifique de chaque bibliothèque. Mettez régulièrement à jour vos dépendances pour résoudre les problèmes de sécurité.

# Versions

Le tableau suivant indique quand les fonctionnalités et les mises à jour logicielles ont été publiées pour le AWS Panorama service, le logiciel et la documentation. Pour vous assurer d'avoir accès à toutes les fonctionnalités, mettez à jour votre AWS Panorama Appliance avec la dernière version logicielle. Pour plus d'informations sur une version, consultez la rubrique associée.

| Modification | Description | Date |
| --- | --- | --- |
| Mise à jour logicielle de l'appliance | La version 7.0.13 est une mise à jour de version majeure qui modifie la façon dont l'appliance gère les mises à jour logicielles. Si vous limitez les communications réseau sortantes depuis l'appliance ou si vous la connectez à un sous-réseau VPC privé, vous devez autoriser l'accès à des points de terminaison et à des ports supplémentaires avant d'appliquer la mise à jour. Pour plus d'informations, consultez le journal des modifications. | 28 décembre 2023 |
| Mise à jour logicielle de l'appliance | La version 6.2.1 inclut des corrections de bugs. Pour plus d'informations, consultez le journal des modifications. | 6 septembre 2023 |
| Mise à jour logicielle de l'appliance | La version 6.0.8 inclut des corrections de bugs et des améliorations de sécurité. Pour plus d'informations, consultez le journal des modifications. | 6 juillet 2023 |

| | | |
|---|---|---|
| Mise à jour logicielle de l'appliance | La version 5.1.7 inclut des corrections de bogues et des améliorations de la gestion des erreurs. Pour plus d'informations, consultez le journal des modifications. | 31 mars 2023 |
| Mise à jour de console | Vous pouvez désormais acheter l'AWS Panoramaappliance depuis la console de gestion. Pour accorder à un utilisateur l'autorisation d'acheter des appareils, consultez les politiques IAM basées sur l'identité pour AWS Panorama. | 2 février 2023 |
| Mise à jour logicielle de l'appliance | La version 5.0.74 inclut des corrections de bugs et des améliorations de gestion des erreurs. Pour plus d'informations, consultez le journal des modifications. | 23 janvier 2023 |
| Mise à jour d'API | Ajout d'`AllowMajorVersionUpdate` une option permettant `OTAJobConfig` d'activer les mises à jour des versions majeures du logiciel de l'appliance. Pour plus d'informations, consultez CreateJobForDevices. | 19 janvier 2023 |

| | | |
|---|---|---|
| Nouvel outil pour les développeurs | Un nouvel outil, le « sideloading », est disponible dans le référentiel d'AWS Panoramaé chantillons GitHub . Vous pouvez utiliser cet outil pour mettre à jour le code d'une application sans créer ni déployer de conteneur. Pour plus d'informations, consultez le fichier README. | 16 novembre 2022 |
| Mise à jour de l'image de base de | La version 1.2.0 ajoute une option de délai d'attente à `video_in.get()` , définit la variable d'`AWS_REGION` environnement et améliore la gestion des erreurs. Pour plus d'informations, consultez le journal des modifications. | 16 novembre 2022 |
| Mise à jour logicielle de l'appliance | La version 5.0.42 inclut des corrections de bugs et des mises à jour de sécurité. Pour plus d'informations, consultez le journal des modifications. | 16 novembre 2022 |
| Mise à jour logicielle de l'appliance | La version 5.0.7 permet de redémarrer les appareils à distance et de suspendre les flux de caméras à distance. Pour plus d'informations, consultez le journal des modifications. | 13 octobre 2022 |

| | | |
|---|---|---|
| Mise à jour logicielle de l'appliance | La version 4.3.93 ajoute la prise en charge de la récupération des journaux depuis un appareil hors ligne. Pour plus d'informations, consultez le journal des modifications. | 24 août 2022 |
| Mise à jour logicielle de l'appliance | La version 4.3.72 inclut des corrections de bugs et des mises à jour de sécurité. Pour plus d'informations, consultez le journal des modifications. | 23 juin 2022 |
| AWS PrivateLink Prise en charge de | AWS Panoramaprend en charge les points de terminaison VPC pour gérer les AWS Panorama ressources à partir d'un sous-réseau privé. Pour plus d'informations, consultez la section Utilisation des points de terminaison VPC. | 2 juin 2022 |
| Mise à jour logicielle de l'appliance | La version 4.3.55 améliore l'utilisation du stockage pour le `console_output` journal. Pour plus d'informations, consultez le journal des modifications. | 5 mai 2022 |

| | | |
|---|---|---|
| Lenovo ThinkEdge® SE70 | Un nouvel appareil pour AWS Panorama est disponible auprès de Lenovo. Le Lenovo ThinkEdge® SE70, alimenté par Nvidia Jetson Xavier NX, prend en charge les mêmes fonctionnalités que l'applian ce. AWS Panorama Pour plus d'informations, consultez la section Appareils compatibles. | 6 avril 2022 |
| Mise à jour de l'image de base de | La version 1.1.0 améliore les performances lors de l'exécution de threads d'arrière -plan et ajoute un indicateu r (is_cached) aux objets multimédia qui indique si l'image est fraîche. Pour plus d'informations, consultez gallery.ecr.aws. | 29 mars 2022 |
| Mise à jour logicielle de l'appliance | La version 4.3.45 ajoute la prise en charge de l'accès au GPU et des ports entrants. Pour plus d'informations, consultez le journal des modifications. | 24 mars 2022 |
| Mise à jour logicielle de l'appliance | La version 4.3.35 améliore la sécurité et les performan ces. Pour plus d'informa tions, consultez le journal des modifications. | 22 février 2022 |

| | | |
|---|---|---|
| Politiques gérées mises à jour | AWS Identity and Access Managementles politiques gérées pour AWS Panorama ont été mises à jour. Pour plus de détails, consultez les politiques gérées par AWS. | 13 janvier 2022 |
| Journaux de provisionnement | Avec le logiciel 4.3.23 de l'appliance, l'appliance écrit des journaux sur une clé USB pendant le provisionnement. Pour plus d'informations, consultez la section Logs. | 13 janvier 2022 |
| Configuration du serveur NTP | Vous pouvez désormais configurer l'AWS Panoramaa ppliance pour utiliser un serveur NTP spécifique pour la synchronisation des horloges. Configurez les paramètres NTP lors de la configuration de l'appliance avec d'autres paramètres réseau. Pour plus d'informations, consultez la section Configuration. | 13 janvier 2022 |
| Régions supplémentaires | AWS Panoramaest désormais disponible dans les régions Asie-Pacifique (Singapour) et Asie-Pacifique (Sydney). | 13 janvier 2022 |

| | | |
|---|---|---|
| Mise à jour logicielle de l'appliance | La version 4.3.4 ajoute le support pour le `precision Mode` paramétrage des modèles et met à jour le comportement de journalisation. Pour plus d'informations, consultez le journal des modifications. | 8 novembre 2021 |
| Politiques gérées mises à jour | AWS Identity and Access Managementles politiques gérées pour AWS Panorama ont été mises à jour. Pour plus de détails, consultez les politiques gérées par AWS. | 20 octobre 2021 |
| Disponibilité générale | AWS Panoramaest désormais disponible pour tous les clients des régions des États-Unis est (Virginie du Nord), de l'ouest des États-Unis (Oregon), de l'Europe (Irlande) et du Canada (centre). Pour acheter un AWS Panorama appareil, rendez-vous sur AWS Panorama. | 20 octobre 2021 |
| Version préliminaire | AWS Panoramaest disponible sur invitation dans les régions USA Est (Virginie du Nord) et USA Ouest (Oregon). | 1er décembre 2020 |

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.