Entwicklerhandbuch

AWS Panorama

AWS Panorama: Entwicklerhandbuch

Table of Contents

Was ist AWS Panorama?

AWS Panoramaist ein Dienst, der Computer Vision in Ihr lokales Kameranetzwerk bringt. Sie installieren dasAWS PanoramaAppliance oder ein anderes kompatibles Gerät in Ihrem Rechenzentrum, registrieren Sie es beiAWS Panorama, und stellen Sie Computer-Vision-Anwendungen aus der Cloud bereit.AWS Panoramafunktioniert mit Ihren vorhandenen Echtzeit-Streaming-Protokoll (RTSP) -Netzwerkkameras. Die Appliance führt sichere Computer Vision-Anwendungen ausAWSPartneroder Anwendungen, die Sie selbst mit demAWS PanoramaAnwendungs-SDK.

DieAWS PanoramaAppliance ist eine kompakte Edge-Appliance, die eine leistungsstarke system-on-module (SOM), das für Arbeitslasten für maschinelles Lernen optimiert ist. Die Appliance kann mehrere Computer Vision-Modelle für mehrere Videostreams parallel ausführen und die Ergebnisse in Echtzeit ausgeben. Es ist für den Einsatz in gewerblichen und industriellen Umgebungen konzipiert und für den Staub- und Flüssigkeitsschutz (IP-62) ausgelegt.

DieAWS PanoramaMit der Appliance können Sie eigenständige Computer Vision-Anwendungen am Netzwerkrand ausführen, ohne Bilder an die AWS-Cloud senden zu müssen. Durch die Verwendung des AWS SDK können Sie sich in andere AWS-Services integrieren und diese verwenden, um Daten aus der Anwendung im Laufe der Zeit zu verfolgen. Durch die Integration von mit anderen AWS-Services können Sie Folgendes verwendenAWS Panoramawie folgt:

- Analysieren Sie Verkehrsmuster— Verwenden Sie das AWS SDK, um Daten für Einzelhandelsanalysen in Amazon DynamoDB aufzuzeichnen. Verwenden Sie eine serverlose Anwendung, um die gesammelten Daten im Laufe der Zeit zu analysieren, Anomalien in den Daten zu erkennen und future Verhalten vorherzusagen.
- Erhalten Sie Sicherheitswarnungen vor— Überwachung von Sperrgebieten an einem Industriestandort. Wenn Ihre Anwendung eine potenziell unsichere Situation erkennt, laden Sie ein Bild in Amazon Simple Storage Service (Amazon S3) hoch und senden Sie eine Benachrichtigung an ein Amazon Simple Notification Service (Amazon SNS) -Thema, damit die Empfänger Korrekturmaßnahmen ergreifen können.
- Verbessern Sie die Qualitätskontrolle— Überwachen Sie die Ausgabe einer Montagelinie, um Teile zu identifizieren, die nicht den Anforderungen entsprechen. Markieren Sie Bilder von nicht konformen Teilen mit Text und einem Begrenzungsrahmen, und zeigen Sie sie zur Überprüfung durch Ihr Qualitätskontrollteam auf einem Monitor an.

- Sammeln Sie Trainings- und Testdaten— Laden Sie Bilder von Objekten hoch, die Ihr Computer Vision-Modell nicht identifizieren konnte oder bei denen das Vertrauen des Modells in seine Vermutung grenzwertig war. Verwenden Sie eine serverlose Anwendung, um eine Warteschlange mit Bildern zu erstellen, die mit Tags versehen werden müssen. Kennzeichnen Sie die Bilder und verwenden Sie sie, um das Modell in Amazon neu zu trainieren SageMakeraus.

AWS Panoramaverwendet andere AWS-Services zur Verwaltung derAWS PanoramaAppliance, greifen Sie auf Modelle und Code zu und stellen Sie Anwendungen bereit.AWS Panoramatut so viel wie möglich, ohne dass Sie mit anderen Diensten interagieren müssen, aber wenn Sie die folgenden Dienste kennen, können Sie verstehen, wieAWS Panoramafunktioniert.

- SageMaker— Sie können verwenden SageMaker um Trainingsdaten von Kameras oder Sensoren zu sammeln, ein Modell für maschinelles Lernen zu erstellen und es für Computer Vision zu trainieren.AWS PanoramaVerwendungszwecke SageMaker Neo zur Optimierung von Modellen für die Ausführung auf demAWS PanoramaAppliances.
- Amazon S3— Sie verwenden Amazon S3 S3-Zugriffspunkte, um Anwendungscode, Modelle und Konfigurationsdateien für die Bereitstellung auf einemAWS PanoramaAppliances.
- AWS IoT–AWS PanoramaVerwendungszweckeAWS IoTDienste zur Überwachung des Status derAWS PanoramaAppliance, Softwareupdates verwalten und Anwendungen bereitstellen. Sie müssen nicht verwendenAWS IoTdirekt.

Dies sind Ihre ersten Schritte mitAWS PanoramaAppliances und erfahren Sie mehr über drn Service, fahren Sie mitErste Schritte mit AWS Panoramaaus.

Erste Schritte mit AWS Panorama

Informieren Sie sich zunächst über die Konzepte des Dienstes und die in diesem Handbuch verwendete Terminologie.AWS Panorama Anschließend können Sie dieAWS Panorama Konsole verwenden, um IhreAWS Panorama Appliance zu registrieren und eine Anwendung zu erstellen. In etwa einer Stunde können Sie das Gerät konfigurieren, die Software aktualisieren und eine Beispielanwendung bereitstellen. Um die Tutorials in diesem Abschnitt abzuschließen, verwenden Sie dieAWS Panorama Appliance und eine Kamera, die Video über ein lokales Netzwerk streamt.

> **Note**
>
> Besuchen Sie die AWS PanoramaKonsole, um einAWS Panorama Gerät zu kaufen.

Die AWS PanoramaBeispielanwendung demonstriert die Verwendung vonAWS Panorama Funktionen. Es enthält ein Modell, das mit einem Trainingsmodell trainiert wurde, SageMaker und einen Beispielcode, der dasAWS Panorama Application SDK verwendet, um Inferenz auszuführen und Videos auszugeben. Die Beispielanwendung enthält eineAWS CloudFormation Vorlage und Skripts, die zeigen, wie Entwicklungs- und Bereitstellungsworkflows von der Befehlszeile aus automatisiert werden können.

Die letzten beiden Themen in diesem Kapitel beschreiben die Anforderungen für Modelle und Kameras sowie die Hardwarespezifikationen derAWS Panorama Appliance. Wenn Sie noch kein Gerät und keine Kameras erworben haben oder planen, Ihre eigenen Computer Vision-Modelle zu entwickeln, finden Sie zunächst in diesen Themen weitere Informationen.

Themen

- AWS-Panorama-Konzepte
- Einrichten des AWS Panorama Appliance
- Bereitstellen der AWS Panorama Panorama-Beispielanwendung
- Entwickeln von AWS Panorama Panorama-Anwendungen
- Unterstützte Computer Vision-Modelle und -Kameras
- Spezifikationen zu AWS Panorama Appliances
- Servicekontingente

AWS-Panorama-Konzepte

In AWS Panorama erstellen Sie Computer-Vision-Anwendungen und stellen sie auf der AWS Panorama Appliance oder einem kompatiblen Gerät bereit, um Videostreams von Netzwerkkameras zu analysieren. Sie schreiben Anwendungscode in Python und erstellen Anwendungscontainer mit Docker. Sie verwenden die AWS Panorama Application CLI, um Modelle für maschinelles Lernen lokal oder aus Amazon Simple Storage Service (Amazon S3) zu importieren. Anwendungen verwenden das AWS Panorama Application SDK, um Videoeingaben von einer Kamera zu empfangen und mit einem Modell zu interagieren.

Konzepte

Die AWS-Panorama-Appliance

Die AWS Panorama Appliance ist die Hardware, auf der Ihre Anwendungen ausgeführt werden. Sie verwenden die AWS-Panorama-Konsole, um eine Appliance zu registrieren, ihre Software zu aktualisieren und Anwendungen darauf bereitzustellen. Die Software auf der AWS Panorama Appliance stellt eine Verbindung zu Kamera-Streams her, sendet Videoframes an Ihre Anwendung und zeigt die Videoausgabe auf einem angeschlossenen Display an.

Die AWS Panorama Appliance ist eineEdge-Gerät unterstützt von Nvidia Jetson AGX Xavier. Anstatt Bilder an die zu sendenAWSCloud für die Verarbeitung: Anwendungen werden lokal auf optimierter Hardware ausgeführt. Auf diese Weise können Sie Videos in Echtzeit analysieren und die Ergebnisse lokal verarbeiten. Die Appliance benötigt eine Internetverbindung, um ihren Status zu melden, Protokolle hochzuladen und Softwareupdates und -bereitstellungen durchzuführen.

Weitere Informationen finden Sie unter Verwalten derAWS PanoramaAppliances.

Kompatible Geräte

Zusätzlich zur AWS Panorama Appliance unterstützt AWS Panorama kompatible Geräte vonAWSPartner. Kompatible Geräte unterstützen dieselben Funktionen wie die AWS Panorama

Appliance. Sie registrieren und verwalten kompatible Geräte mit der AWS-Panorama-Konsole und der API und erstellen und implementieren Anwendungen auf die gleiche Weise.

- Lenovo ThinkEdge® SE 70— Bereitgestellt von Nvidia Jetson Xavier NX

Der Inhalt und die Beispielanwendungen in diesem Handbuch wurden mit der AWS Panorama Appliance entwickelt. Weitere Informationen zu bestimmten Hardware- und Softwarefunktionen für Ihr Gerät finden Sie in der Dokumentation des Herstellers.

Anwendungen

Anwendungen werden auf der AWS Panorama Appliance ausgeführt, um Computer-Vision-Aufgaben in Videostreams auszuführen. Sie können Computer-Vision-Anwendungen erstellen, indem Sie Python-Code und Modelle für maschinelles Lernen kombinieren und sie über das Internet auf der AWS Panorama Appliance bereitstellen. Anwendungen können Videos an ein Display senden oder das AWS-SDK verwenden, um Ergebnisse an AWS-Services zu senden.

Um Anwendungen zu erstellen und bereitzustellen, verwenden Sie die AWS Panorama Application CLI. Die AWS Panorama Application CLI ist ein Befehlszeilentool, das Standardanwendungsordner und Konfigurationsdateien generiert, Container mit Docker erstellt und Ressourcen hochlädt. Sie können mehrere Anwendungen auf einem Gerät ausführen.

Weitere Informationen finden Sie unter Verwalten vonAWS PanoramaAnwendungen.

Knoten

Eine Anwendung besteht aus mehreren Komponenten namensKnoten, die Eingaben, Ausgaben, Modelle und Code darstellen. Ein Knoten kann nur konfigurativ sein (Eingaben und Ausgaben) oder Artefakte (Modelle und Code) enthalten. Die Codeknoten einer Anwendung sind gebündeltKnoten-Paketedie Sie auf einen Amazon S3-Zugriffspunkt hochladen, wo die AWS Panorama Appliance auf sie zugreifen kann. EinAnwendungsmanifestist eine Konfigurationsdatei, die Verbindungen zwischen den Knoten definiert.

Weitere Informationen finden Sie unter Anwendungs-Knoten.

Modelle

Ein Computer-Vision-Modell ist ein Netzwerk für maschinelles Lernen, das darauf trainiert ist, Bilder zu verarbeiten. Computer-Vision-Modelle können verschiedene Aufgaben wie Klassifizierung,

Erkennung, Segmentierung und Verfolgung ausführen. Ein Computer-Vision-Modell verwendet ein Bild als Eingabe und gibt Informationen über das Bild oder die Objekte im Bild aus.

AWS Panorama unterstützt Modelle, die mit erstellt wurden PyTorch, Apache MXNet und TensorFlow. Sie können Modelle mit Amazon erstellen SageMaker oder in Ihrer Entwicklungsumgebung. Weitere Informationen finden Sie unter ???.

Einrichten des AWS Panorama Appliance

Um mit der Nutzung Ihrer AWS Panorama Panorama-Appliance oder Ihres kompatiblen Geräts zu beginnen, registrieren Sie es in der AWS Panorama Panorama-Konsole und aktualisieren Sie die zugehörige Software. Während des Einrichtungsvorgangs erstellen Sie in AWS Panorama eine Appliance-Ressource, die die physische Appliance darstellt, und kopieren Dateien mit einem USB-Laufwerk auf die Appliance. Die Appliance verwendet diese Zertifikate und Konfigurationsdateien, um eine Verbindung zum AWS Panorama Panorama-Service herzustellen. Anschließend verwenden Sie die AWS-Panorama-Konsole, um die Software der Appliance zu aktualisieren und Kameras zu registrieren.

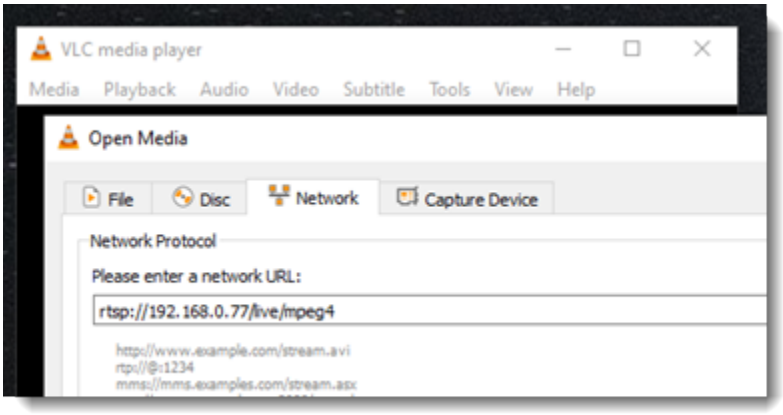
Abschnitte

Voraussetzungen

Damit Sie die Schritte in diesem Tutorial ausführen können, müssen Sie über eine AWS Panorama Appliance oder ein kompatibles Gerät und die folgende Hardware verfügen:

- Display — Ein Display mit HDMI-Eingang zur Anzeige der Ausgabe der Beispielanwendung.
- USB-Laufwerk (im Lieferumfang der AWS Panorama Appliance enthalten) — Ein FAT32-formatiertes USB 3.0-Flash-Speicherlaufwerk mit mindestens 1 GB Speicherplatz für die Übertragung eines Archivs mit Konfigurationsdateien und einem Zertifikat auf die AWS Panorama Appliance.
- Kamera — Eine IP-Kamera, die einen RTSP-Videostream ausgibt.

Verwenden Sie die Tools und Anweisungen des Herstellers Ihrer Kamera, um die IP-Adresse und den Stream-Pfad der Kamera zu ermitteln. Sie können einen Videoplayer wie VLC verwenden, um die Stream-URL zu überprüfen, indem Sie ihn als Netzwerk-Medienquelle öffnen:



Die AWS-Panorama-Konsole verwendet andere AWS-Services, um Anwendungskomponenten zusammenzustellen, Berechtigungen zu verwalten und Einstellungen zu überprüfen. Um eine Appliance zu registrieren und die Beispielanwendung bereitzustellen, benötigen Sie die folgenden Berechtigungen:

- AWSPanoramaFullAccess— Bietet vollen Zugriff auf AWS Panorama, AWS Panorama-Zugangspunkte in Amazon S3, Appliance-Anmeldeinformationen und Appliance-Logs in AmazonCloudWatch. AWS Secrets Manager Beinhaltet die Erlaubnis, eine serviceverknüpfte Rolle für AWS Panorama zu erstellen.
- AWS Identity and Access Management(IAM) — Beim ersten Durchlauf, um Rollen zu erstellen, die vom AWS Panorama-Service und der AWS Panorama Appliance verwendet werden.

Wenn Sie nicht berechtigt sind, Rollen in IAM zu erstellen, bitten Sie einen Administrator, die AWS-Panorama-Konsole zu öffnen und die Aufforderung zur Erstellung von Servicerollen zu akzeptieren.

Registrieren und konfigurieren Sie die AWS Panorama Appliance

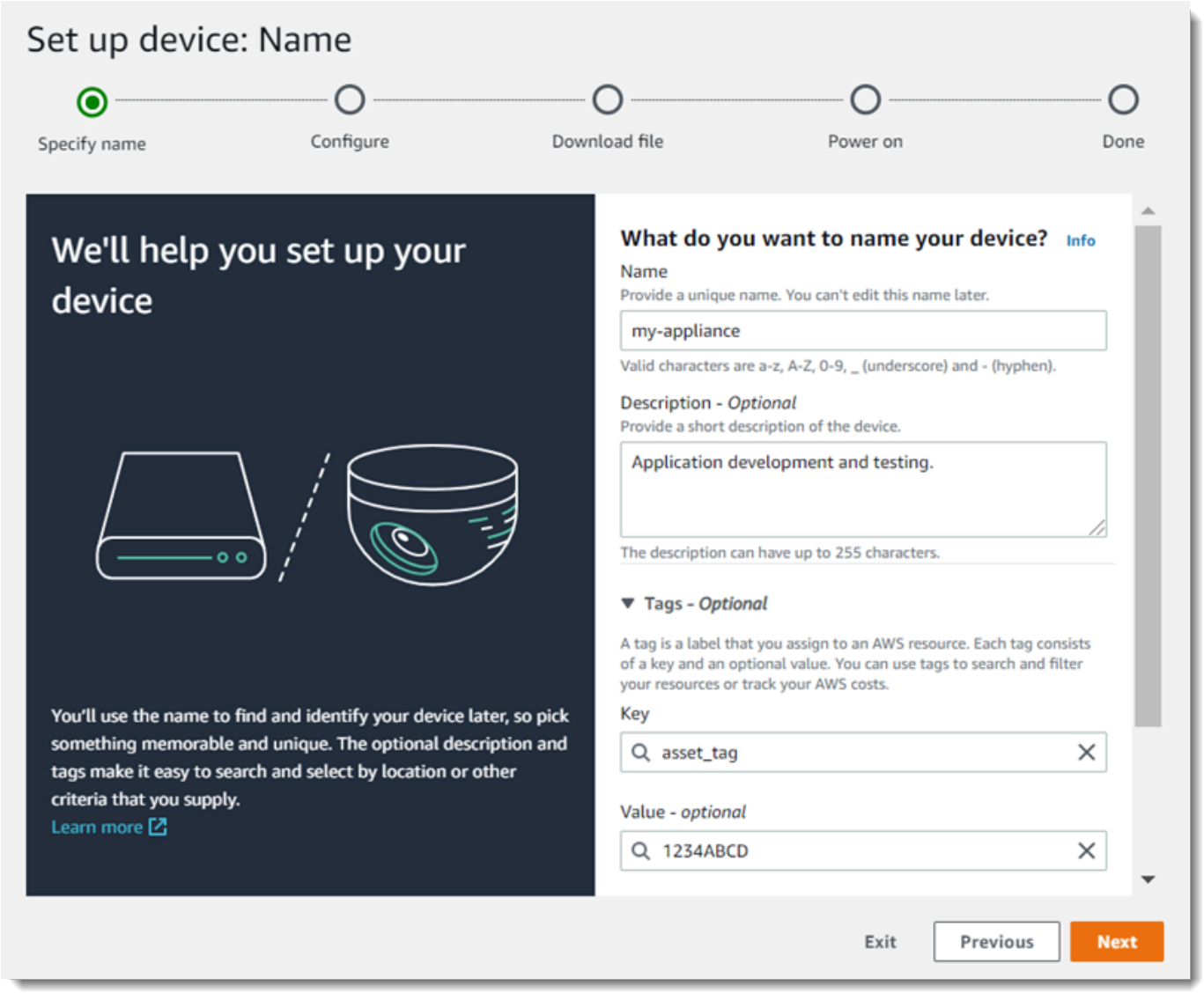
Die AWS Panorama Appliance ist ein Hardwaregerät, das über eine lokale Netzwerkverbindung eine Verbindung zu netzwerkfähigen Kameras herstellt. Es verwendet ein Linux-basiertes Betriebssystem, das das AWS Panorama Application SDK und unterstützende Software für die Ausführung von Computer-Vision-Anwendungen enthält.

Um eine Verbindung AWS für die Appliance-Verwaltung und Anwendungsbereitstellung herzustellen, verwendet die Appliance ein Gerätezertifikat. Sie verwenden die AWS Panorama Panorama-Konsole, um ein Bereitstellungszertifikat zu generieren. Die Appliance verwendet dieses temporäre Zertifikat, um die Ersteinrichtung abzuschließen und ein permanentes Gerätezertifikat herunterzuladen.

> **Important**
>
> Das Bereitstellungszertifikat, das Sie in diesem Verfahren generieren, ist nur 5 Minuten gültig. Wenn Sie den Registrierungsprozess nicht innerhalb dieses Zeitraums abschließen, müssen Sie von vorne beginnen.

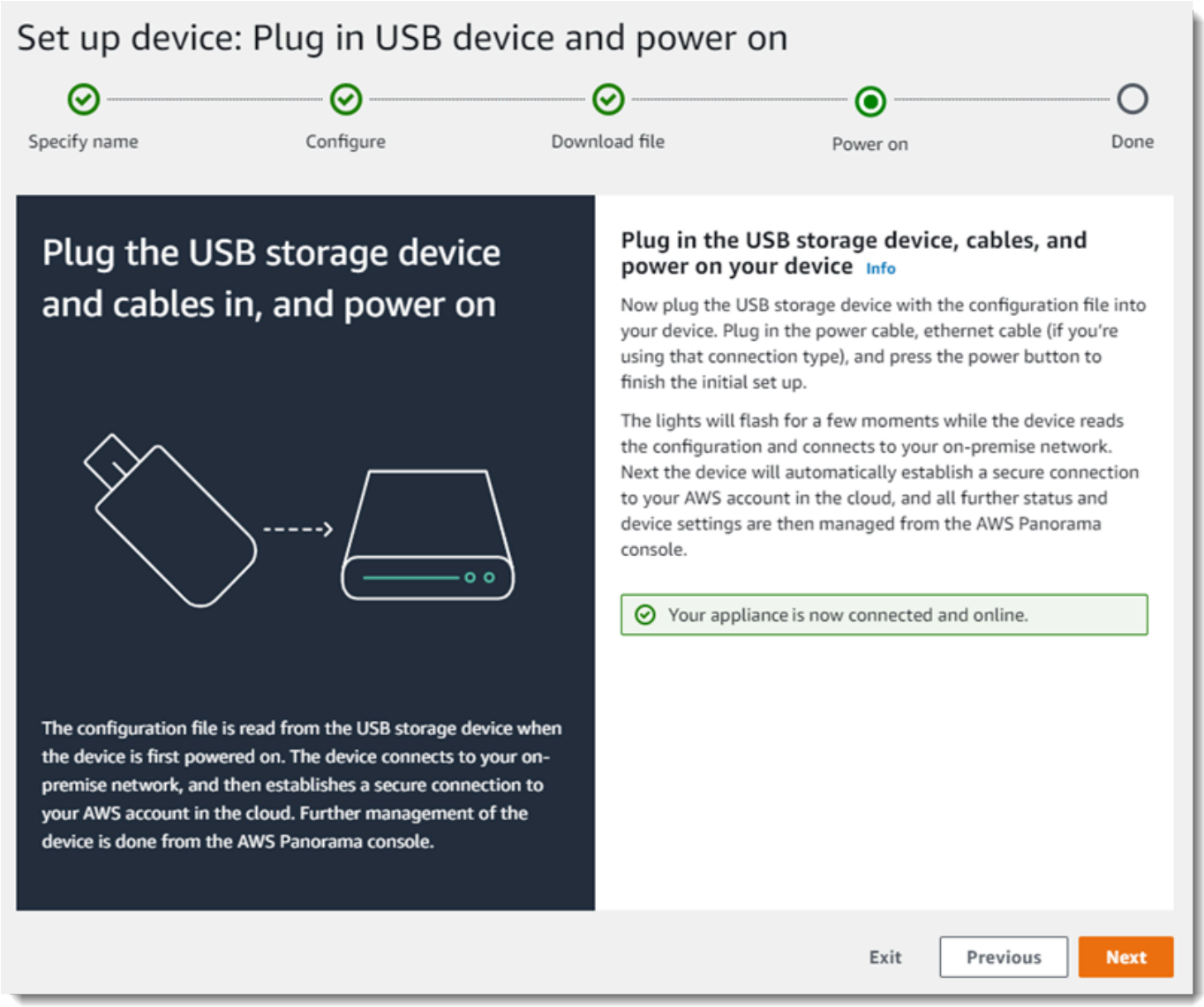
Um ein Gerät zu registrieren

1. Connect Sie das USB-Laufwerk mit Ihrem Computer. Bereiten Sie das Gerät vor, indem Sie die Netzwerk- und Stromkabel anschließen. Die Appliance wird eingeschaltet und wartet darauf, dass ein USB-Laufwerk angeschlossen wird.
2. Öffnen Sie die Seite Erste Schritte der AWS Panorama Panorama-Konsole.
3. Wählen Sie Gerät hinzufügen.
4. Wählen Sie „Einrichtung starten“.
5. Geben Sie einen Namen und eine Beschreibung für die Geräteressource ein, die die Appliance in AWS Panorama darstellt. Wählen Sie Next (Weiter)



6. Wenn Sie manuell eine IP-Adresse, einen NTP-Server oder DNS-Einstellungen zuweisen müssen, wählen Sie Erweiterte Netzwerkeinstellungen. Klicken Sie andernfalls auf Next (Weiter).
7. Wählen Sie Archiv herunterladen. Wählen Sie Weiter.
8. Kopieren Sie das Konfigurationsarchiv in das Stammverzeichnis des USB-Laufwerks.
9. Connect das USB-Laufwerk an den USB 3.0-Anschluss an der Vorderseite des Geräts neben dem HDMI-Anschluss an.

   Wenn Sie das USB-Laufwerk anschließen, kopiert die Appliance das Konfigurationsarchiv und die Netzwerkkonfigurationsdatei in sich selbst und stellt eine Verbindung zur AWS Cloud her. Die Statusanzeige der Appliance leuchtet von grün auf blau, während die Verbindung hergestellt ist, und dann wieder auf grün.
10. Wählen Sie Next, um fortzufahren.



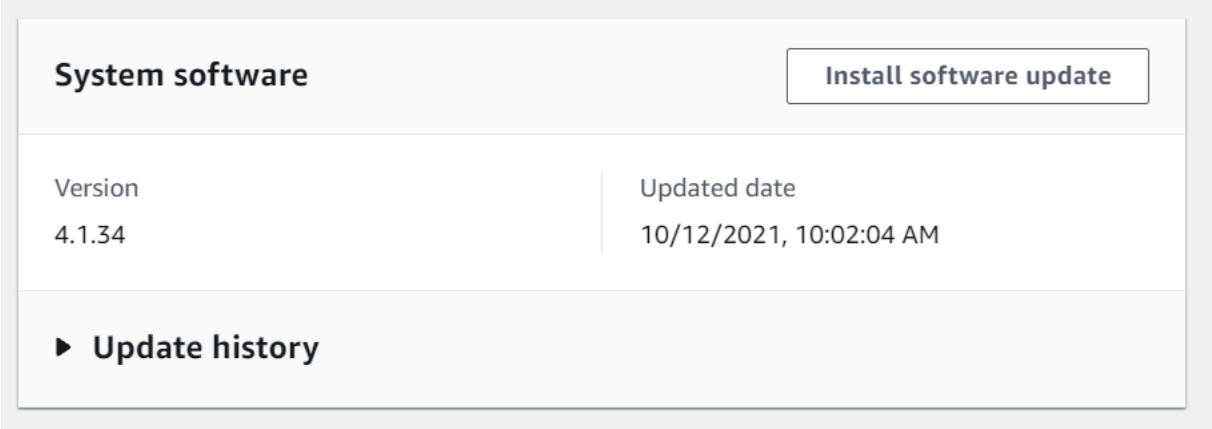
11. Wählen Sie Done (Erledigt) aus.

Aktualisieren Sie die Appliance-Software

Die AWS Panorama Appliance umfasst mehrere Softwarekomponenten, darunter ein Linux-Betriebssystem, das AWS Panorama Panorama-Anwendungs-SDK und unterstützende Bibliotheken und Frameworks für maschinelles Sehen. Um sicherzustellen, dass Sie die neuesten Funktionen und Anwendungen mit Ihrer Appliance verwenden können, aktualisieren Sie die Software nach der Einrichtung und wann immer ein Update verfügbar ist.

Um die Appliance-Software zu aktualisieren

1. Öffnen Sie die Seite Geräte der AWS Panorama Panorama-Konsole.
2. Wählen Sie ein Gerät.
3. Wählen Sie „Einstellungen“
4. Wählen Sie unter Systemsoftware die Option Softwareupdate installieren aus.



5. Wählen Sie eine neue Version und wählen Sie dann Installieren.

> **Important**
>
> Bevor Sie fortfahren, entfernen Sie das USB-Laufwerk aus der Appliance und formatieren Sie es, um seinen Inhalt zu löschen. Das Konfigurationsarchiv enthält sensible Daten und wird nicht automatisch gelöscht.

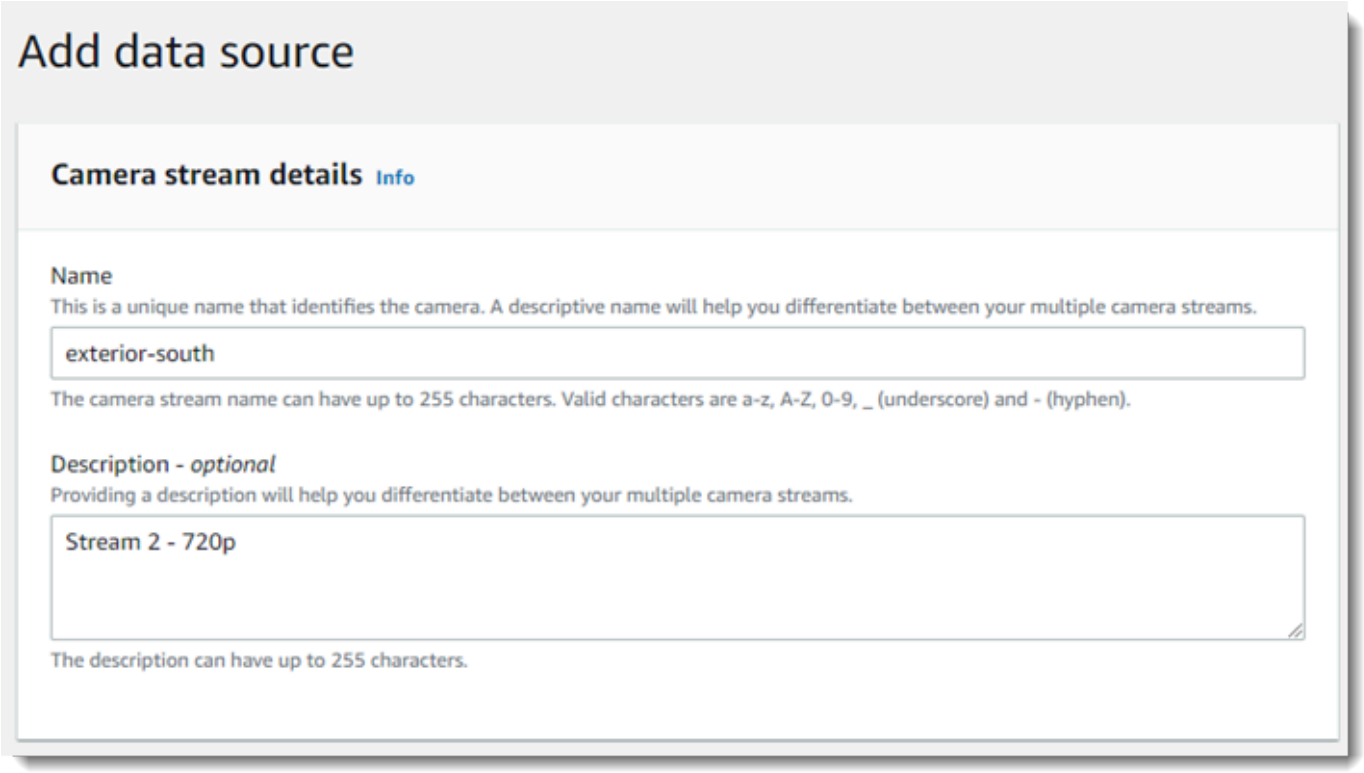
Der Upgrade-Vorgang kann 30 Minuten oder länger dauern. Sie können den Fortschritt in der AWS-Panorama-Konsole oder auf einem angeschlossenen Monitor überwachen. Wenn der Vorgang abgeschlossen ist, wird die Appliance neu gestartet.

Fügen Sie einen Kamerastream hinzu

Registrieren Sie als Nächstes einen Kamerastream bei der AWS-Panorama-Konsole.

So registrieren Sie einen Kamerastream

1. Öffnen Sie die Seite mit den Datenquellen der AWS Panorama Panorama-Konsole.
2. Wählen Sie Datenquelle hinzufügen aus.



3. Konfigurieren Sie die folgenden Einstellungen.

   - Name — Ein Name für den Kamerastream.
   - Beschreibung — Eine kurze Beschreibung der Kamera, ihres Standorts oder anderer Details.
   - RTSP-URL — Eine URL, die die IP-Adresse der Kamera und den Pfad zum Stream angibt. Beispiel: `rtsp://192.168.0.77/live/mpeg4/`
   - Anmeldeinformationen — Damit Sie die Schritte in diesem Video ausführen können, müssen Sie über einen kennwortgeschützten Namen und ein Kennwort verfügen.
4. Wählen Sie Speichern.

AWS Panorama speichert die Anmeldeinformationen Ihrer Kamera sicher inAWS Secrets Manager. Mehrere Anwendungen können denselben Kamerastream gleichzeitig verarbeiten.

Nächste Schritte

Falls Sie bei der Installation auf Fehler gestoßen sind, finden Sie weitere Informationen unterFehlerbehebung.

Um eine Beispielanwendung bereitzustellen, fahren Sie mit dem nächsten Thema fort.

Bereitstellen der AWS Panorama Panorama-Beispielanwendung

Nachdem durichten Sie Ihre AWS Panorama Appliance oder ein kompatibles Gerät einund aktualisierte seine Software, stellen Sie eine Beispielanwendung bereit. In den folgenden Abschnitten importieren Sie eine Beispielanwendung mit der AWS Panorama Application CLI und stellen sie mit der AWS Panorama Panorama-Konsole bereit.

Die Beispielanwendung verwendet ein Machine-Learning-Modell, um Objekte in Frames eines Videos von einer Netzwerkkamera zu klassifizieren. Es verwendet das AWS Panorama Application SDK, um ein Modell zu laden, Images abzurufen und das Modell auszuführen. Die Anwendung überlagert dann die Ergebnisse über das Originalvideo und gibt sie an ein angeschlossenes Display aus.

In einer Einzelhandelsumgebung können Sie durch die Analyse von Fußgängerverkehrsmustern das Verkehrsaufkommen vorhersagen. Durch die Kombination der Analyse mit anderen Daten können Sie den erhöhten Personalbedarf rund um Feiertage und andere Veranstaltungen planen, die Wirksamkeit von Anzeigen und Verkaufsaktionen messen oder die Displayplatzierung und Bestandsverwaltung optimieren.

Abschnitte

Voraussetzungen

Für die Verfahren in diesem Tutorial benötigen Sie ein Befehlszeilen-Terminal oder eine Befehlszeilen-Shell zum Ausführen der Befehle. In den Codelistungen sind ein ein ein ein ein ein ein ein ein das aktuelle Verzeichnis vorangestellt (bei Bedarf).

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

Für lange Befehle verwenden wir ein Escape-Zeichen (\), um einen Befehl über mehrere Zeilen aufzuteilen.

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das Windows-Subsystem für Linux installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten. Hilfe zum Einrichten einer Entwicklungsumgebung in Windows finden Sie unterEinrichten einer Entwicklungsumgebung in Windows.

Sie verwenden Python, um AWS Panorama Panorama-Anwendungen zu entwickeln und Tools mit pip, dem Paketmanager von Python, zu installieren. Wenn Sie Python noch nicht haben,installieren der neuesten Version. Wenn Sie Python 3, aber nicht pip haben, installieren Sie pip mit dem Paketmanager Ihres Betriebssystems oder installieren Sie eine neue Version von Python, die mit pip geliefert wird.

In diesem Lernprogramm verwenden Sie Docker, um den Container zu erstellen, der Ihren Anwendungscode ausführt. Installieren Sie Docker von der Docker-Website: Holen Sie sich Docker

In diesem Lernprogramm wird die AWS Panorama Application CLI verwendet, um die Beispielanwendung zu importieren, Pakete zu erstellen und Artefakte hochzuladen. Die AWS Panorama Application CLI verwendet dieAWS Command Line Interface(AWS CLI), um Service-API-Operationen aufzurufen. Wenn du das schon hastAWS CLI, aktualisiere es auf die neueste Version. So installieren Sie die AWS Panorama Application CLI undAWS CLI, verwendenpip.

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

Laden Sie die Beispielanwendung herunter und extrahieren Sie sie in Ihren Workspace.

- Beispielanwendung–aws-panorama-sample.zip

Importieren der Beispielanwendung

Um die Beispielanwendung zur Verwendung in Ihrem Konto zu importieren, verwenden Sie die AWS Panorama Application CLI. Die Ordner und das Manifest der Anwendung enthalten Verweise auf eine Platzhalterkontonummer. Um diese mit Ihrer Kontonummer zu aktualisieren, führen Sie die`panorama-cli import-application`befehlen.

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

DieSAMPLE_CODEPaket, in derpackagesenthält den Code und die Konfiguration der Anwendung, einschließlich einer Dockerdatei, die das Basisimage der Anwendung verwendet,panorama-application. Um den Anwendungscontainer zu erstellen, der auf der Appliance ausgeführt wird, verwenden Siepanorama-cli build-containerbefehlen.

```
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
```

Der letzte Schritt mit der AWS Panorama Application CLI besteht darin, den Code und die Modellknoten der Anwendung zu registrieren und Assets auf einen vom Service bereitgestellten Amazon S3 S3-Zugriffspunkt hochzuladen. Zu den Assets gehören das Container-Image des Codes, das Modell und jeweils eine Deskriptordatei. Um die Knoten zu registrieren und Assets hochzuladen, führen Siepanorama-cli package-applicationbefehlen.

```
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
```

Bereitstellen der Anwendung

Verwenden Sie die AWS Panorama Panorama-Konsole, um die Anwendung auf Ihrer Appliance bereitzustellen.

So stellen Sie die Anwendung bereit

1. Öffnen Sie die AWS Panorama Panorama-KonsoleSeite „Bereitgestellte Anwendungen.
2. WählenBereitstellen der Anwendung.
3. Fügen Sie den Inhalt des Anwendungsmanifests ein,graphs/aws-panorama-sample/graph.jsonin den Texteditor. Wählen Sie Next (Weiter).
4. Geben Sie für Application name (Anwendungsname) den Text aws-panorama-sample ein.
5. WählenFortfahren mit der Bereitstellung.
6. WählenStarten der Bereitstellung.

7. WählenWeiterohne eine Rolle auszuwählen.
8. WählenWählen Sie ein Gerät, und wählen Sie dann Ihr Gerät aus. Wählen Sie Next (Weiter).
9. Auf derWählen Sie DatenquellenSchritt, wähleEingabe (s) anzeigenund fügen Sie Ihren Kamerastream als Datenquelle hinzu. Wählen Sie Next (Weiter).
10. Auf derKonfigurationSchritt, wähleWeiter.
11. WählenBereitstellenund dann wählenFertig.
12. Wählen Sie in der Liste der bereitgestellten Anwendungenaws-panorama-sample.

Aktualisieren Sie diese Seite für Updates, oder verwenden Sie das folgende Skript, um die Bereitstellung über die Befehlszeile zu überwachen.

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": {}
    }
]
...
```

Wenn die Anwendung nicht gestartet wird, überprüfen Sie dieAnwendungs- und Geräteprotokolle bei Amazon CloudWatch Logs.

Die Konsolenausgabe anzeigen

Wenn die Bereitstellung abgeschlossen ist, beginnt die Anwendung mit der Verarbeitung des Videostreams und sendet Protokolle an CloudWatch.

So zeigen Sie Logs an CloudWatch Logs (Protokolle)

1. Öffnen Sie dasSeite „Log Groups“ der CloudWatch -Konsole.
2. Suchen Sie AWS Panorama Anwendungs- und Appliance-Protokolle in den folgenden Gruppen:

   - Geräteprotokolle–`/aws/panorama/devices/`*`device-id`*
   - Anwendungs-Logs–`/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
```

Um die Videoausgabe der Anwendung anzuzeigen, schließen Sie das Gerät mit einem HDMI-Kabel an einen Monitor an. Standardmäßig zeigt die Anwendung jedes Klassifizierungsergebnis mit einer Konfidenz von mehr als 20% an.

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",
...
```

Das Beispielmodell umfasst 1000 Klassen, darunter viele Tiere, Lebensmittel und gemeinsame Objekte. Richten Sie Ihre Kamera auf eine Tastatur oder eine Kaffeetasse.

Der Einfachheit halber verwendet die Beispielanwendung ein leichtgewichtiges Klassifizierungsmodell. Das Modell gibt ein einzelnes Array mit einer Wahrscheinlichkeit für jede seiner Klassen aus. Reale Anwendungen verwenden häufiger Objekterkennungsmodelle mit mehrdimensionaler Ausgabe. Beispielanwendungen mit komplexeren Modellen finden Sie unterBeispielanwendungen, Skripte und Vorlagen.

Aktivieren des SDK für Python

Die Beispielanwendung verwendet dasAWS SDK for Python (Boto)um Metriken an Amazon zu senden CloudWatch. Um diese Funktionalität zu aktivieren, erstellen Sie eine Rolle, die der Anwendung die Berechtigung zum Senden von Metriken erteilt, und stellen Sie die Anwendung mit der zugewiesenen Rolle erneut bereit.

Die Beispielanwendung umfasst einAWS CloudFormationVorlage, die eine Rolle mit den erforderlichen Berechtigungen erstellt. Um die Rolle anzulegen, verwenden Sie das`aws cloudformation deploy`befehlen.

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

So stellen Sie die Anwendung erneut bereit

1. Öffnen Sie die AWS Panorama Panorama-KonsoleSeite „Bereitgestellte Anwendungen.
2. Wählen Sie eine Anwendung aus.
3. Wählen Sie Replace (Ersetzen) aus.
4. Führen Sie die Schritte zum Bereitstellen der Anwendung aus. In derAuswählen der IAM-Rolle, wählen Sie die Rolle aus, die Sie erstellt haben. Ihr Name beginnt mit `aws-panorama-sample-runtime`.
5. Wenn die Bereitstellung abgeschlossen ist, öffnen Sie dieCloudWatchKonsoleund sehen Sie sich die Metriken im`AWSPanoramaApplication`-Namespace Alle 150 Frames protokolliert und lädt die Anwendung Metriken für die Frame-Verarbeitung und die Inferenzzeit hoch.

Bereinigen

Wenn Sie mit der Arbeit mit der Beispielanwendung fertig sind, können Sie sie mithilfe der AWS Panorama Panorama-Konsole aus der Appliance entfernen.

So entfernen Sie die Anwendung von der Appliance

1. Öffnen Sie die AWS Panorama Panorama-KonsoleSeite „Bereitgestellte Anwendungen.
2. Wählen Sie eine Anwendung aus.
3. WählenLöschen vom Gerät.

Nächste Schritte

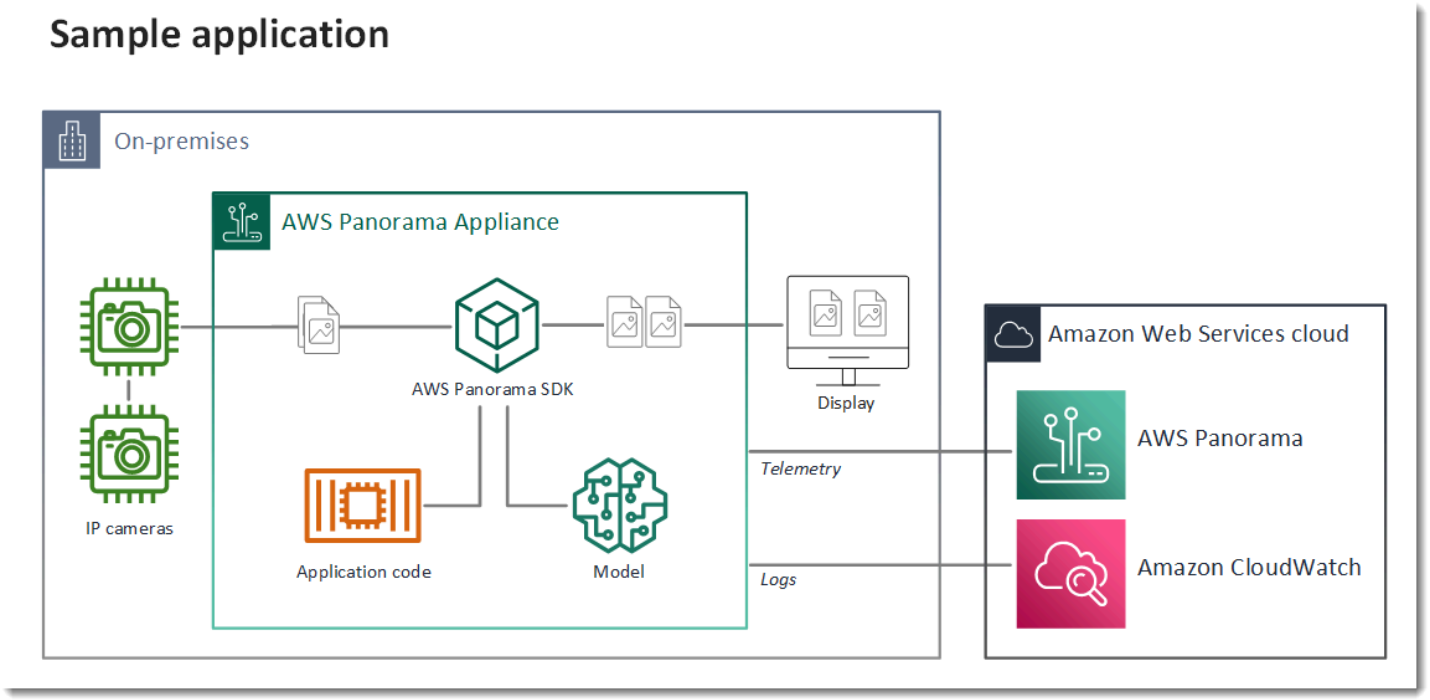
Wenn beim Bereitstellen oder Ausführen der Beispielanwendung Fehler aufgetreten sind, finden Sie weitere Informationen unterFehlerbehebung.

Weitere Informationen zu den Funktionen und der Implementierung der Beispielanwendung finden Sie unterdas nächste Thema.

Entwickeln von AWS Panorama Panorama-Anwendungen

Nutzen Sie die Beispielanwendung, um mehr über die AWS Panorama Panorama-Anwendungsstruktur zu erfahren, und als Ausgangspunkt für Ihre eigene App.

Das folgende Diagramm zeigt die Hauptkomponenten der Anwendung, die auf einer AWS Panorama Appliance ausgeführt wird. Der Anwendungscode verwendet das AWS Panorama Application SDK, um Bilder abzurufen und mit dem Modell zu interagieren, auf das er keinen direkten Zugriff hat. Die Anwendung gibt Video an ein angeschlossenes Display aus, sendet jedoch keine Bilddaten außerhalb Ihres lokalen Netzwerks.



In diesem Beispiel verwendet die Anwendung das AWS Panorama Application SDK, um Videobilder von einer Kamera abzurufen, die Videodaten vorzuverarbeiten und die Daten an ein Computer Vision-Modell zu senden, das Objekte erkennt. Die Anwendung zeigt das Ergebnis auf einem an das Gerät angeschlossenen HDMI-Display an.

Abschnitte

- Hochladen von Metriken mit dem SDK für Python
- Nächste Schritte

Das Anwendungsmanifest

Das Anwendungsmanifest ist eine Datei mit dem Namen`graph.json`im`graphs`folder. Das Manifest definiert die Komponenten der Anwendung, die Pakete, Knoten und Kanten sind.

Pakete sind Code-, Konfigurations- und Binärdateien für Anwendungscode, Modelle, Kameras und Displays. Die Beispielanwendung verwendet 4 Pakete:

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

```
    "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"
        }
    ],
```

Die ersten beiden Pakete sind innerhalb der Anwendung definiert, in der`packages`-Verzeichnis. Sie enthalten den Code und das Modell, die für diese Anwendung spezifisch sind. Die zweiten beiden Pakete sind generische Kamera- und Display-Pakete, die vom AWS Panorama Panorama-Service bereitgestellt werden. Die`abstract_rtsp_media_source`-Paket ist ein Platzhalter für eine Kamera, die Sie während der Bereitstellung überschreiben. Die`hdmi_data_sink`-Paket stellt den HDMI-Ausgangsanschluss am Gerät dar.

Knoten sind Schnittstellen zu Paketen sowie Nicht-Paket-Parameter, die Standardwerte haben können, die Sie bei der Bereitstellung außer Kraft setzen. Die Code- und Modellpakete definieren

Schnittstellen in`package.json`Dateien, die Ein- und Ausgänge angeben, bei denen es sich um Videostreams oder um einen Basisdatentyp wie Float, Boolean oder String handeln kann.

Zum Beispiel, das`code_node`-Knoten bezieht sich auf eine Schnittstelle aus dem`SAMPLE_CODE`Paket.

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

Diese Schnittstelle wird in der Paketkonfigurationsdatei definiert`package.json`aus. Die Schnittstelle gibt an, dass das Paket Geschäftslogik ist und dass es einen Videostream mit dem Namen`video_in`und eine Fließkommazahl namens`threshold`als Eingaben. Die Schnittstelle gibt auch an, dass der Code einen Videostream-Puffer mit dem Namen`video_out`um Video auf einem Display auszugeben

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"
                    }
                ]
            }
        ]
    }
}
```

Zurück im Anwendungsmanifest wurde das`camera_node`-Knoten repräsentiert einen Videostream von einer Kamera. Es enthält einen Decorator, der in der Konsole angezeigt wird, wenn Sie die Anwendung bereitstellen und Sie auffordert, einen Kamerastream auszuwählen.

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

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

Ein Parameterknoten,`threshold_param`definiert den Konfidenzschwellenparameter, der vom Anwendungscode verwendet wird. Sie hat einen Standardwert von 60 und kann während der Bereitstellung außer Kraft gesetzt werden.

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

```
            {
                "name": "threshold_param",
                "interface": "float32",
                "value": 60.0,
                "overridable": true,
```

```
                "decorator": {
                    "title": "Confidence threshold",
                    "description": "The minimum confidence for a classification to be
recorded."
                }
            }
```

Der letzte Abschnitt des Anwendungsmanifests,`edges`, stellt Verbindungen zwischen Knoten her. Der Videostream der Kamera und der Schwellenwertparameter werden mit dem Eingang des Codeknotens verbunden, und der Videoausgang vom Codeknoten wird mit dem Display verbunden.

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

```
        "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"
            }
        ]
```

Erstellen mit der Beispielanwendung

Sie können die Beispielanwendung als Ausgangspunkt für Ihre eigene App verwenden.

Der Name jedes Pakets muss in Ihrem Konto eindeutig sein. Wenn Sie und ein anderer Benutzer in Ihrem Konto beide einen generischen Paketnamen wie`codeodermodel`verwenden, erhalten Sie möglicherweise die falsche Version des Pakets, wenn Sie es bereitstellen. Ändern Sie den Namen des Code-Pakets in einen Namen, der Ihre Anwendung darstellt.

So benennen Sie das Codepaket um

1. Benennen Sie den Paketordner um:`packages/123456789012-`*`SAMPLE_CODE`*`-1.0/`aus.
2. Aktualisieren Sie den Paketnamen an den folgenden Speicherorten.

- AnwendungsManifest–`graphs/aws-panorama-sample/graph.json`
- Paketkonfiguration–`packages/123456789012-SAMPLE_CODE-1.0/package.json`
- Build-Skript–`3-build-container.sh`

So aktualisieren Sie den Anwendungscode

1. Ändern Sie den Anwendungscode in`packages/123456789012-SAMPLE_CODE-1.0/src/application.py`aus.
2. Führen Sie zum Erstellen des Containers`3-build-container.sh`aus.

```
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
```

Die CLI löscht automatisch das alte Container-Asset aus dem`assets`und aktualisiert die Paketkonfiguration.

3. Führen Sie aus, um die Pakete hochzuladen`4-package-application.py`aus.
4. Öffnen Sie die AWS Panorama Panorama-KonsoleBereitgestellte Anwendungsseiteaus.
5. Wählen Sie eine Anwendung aus.
6. Wählen Sie Replace (Ersetzen) aus.
7. Führen Sie die Schritte aus, um die Anwendung bereitzustellen. Bei Bedarf können Sie Änderungen am Anwendungsmanifest, an Kamerastreams oder an Parametern vornehmen.

Änderung des Computer Vision-Modells

Die Beispielanwendung enthält ein Computer Vision-Modell. Um Ihr eigenes Modell zu verwenden, ändern Sie die Konfiguration des Modellknotens und verwenden Sie die AWS Panorama Application CLI, um ihn als Asset zu importieren.

Das folgende Beispiel verwendet eine MxNet-SSD ResNet50 Modelle, die Sie in diesem Handbuch herunterladen können GitHub repo:ssd_512_resnet50_v1_voc.tar.gz

So ändern Sie das Modell der Beispielanwendung

1. Benennen Sie den Paketordner um, damit er Ihrem Modell entspricht Zum Beispiel, um`packages/`*`123456789012`*`-`*`SSD_512_RESNET50_V1_VOC`*`-1.0/`aus.
2. Aktualisieren Sie den Paketnamen an den folgenden Speicherorten.

   - AnwendungsManifest–`graphs/aws-panorama-sample/graph.json`
   - Paketkonfiguration–`packages/`*`123456789012`*`-`*`SSD_512_RESNET50_V1_VOC`*`-1.0/package.json`
3. In der Paketkonfigurationsdatei (`package.json`) enthalten. Ändern Sie die`assets`Wert in ein leeres Array.

```
{
    "nodePackage": {
        "envelopeVersion": "2021-01-01",
        "name": "SSD_512_RESNET50_V1_VOC",
        "version": "1.0",
        "description": "Compact classification model",
```

```
        "assets": [],
```

4. Öffnen Sie die Paketdeskriptordatei (`descriptor.json`) enthalten. Aktualisieren des`framework`und`shape`Werte, die zu Ihrem Modell passen.

```
{
    "mlModelDescriptor": {
        "envelopeVersion": "2021-01-01",
        "framework": "MXNET",
        "inputs": [
            {
                "name": "data",
                "shape": [ 1, 3, 512, 512 ]
            }
        ]
    }
}
```

   Der Wert für`Form`,`1,3,512,512`, gibt die Anzahl der Bilder an, die das Modell als Eingabe verwendet (1), die Anzahl der Kanäle in jedem Bild (3 — Rot, Grün und Blau) und die Abmessungen des Bildes (512 x 512). Die Werte und die Reihenfolge des Arrays variieren je nach Modell.

5. Importieren Sie das Modell mit der AWS Panorama Panorama-Anwendungs-CLI. Die AWS Panorama Application CLI kopiert die Modell- und Deskriptordateien in die`assets`Ordner mit eindeutigen Namen und aktualisiert die Paketkonfiguration.

```
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. Um das Modell hochzuladen, führen Sie`panorama-cli package-application`aus.

```
$ 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. Aktualisieren Sie den Anwendungscode. Der größte Teil des Codes kann wiederverwendet werden. Der Code, der für die Antwort des Modells spezifisch ist, befindet sich im`process_results`-Methode.

```
    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)
```

Abhängig von Ihrem Modell müssen Sie möglicherweise auch die`preprocess`-Methode.

Vorverarbeitung von Bildern

Bevor die Anwendung ein Bild an das Modell sendet, bereitet sie es für die Inferenz vor, indem sie die Größe ändert und die Farbdaten normalisiert. Das Modell, das die Anwendung verwendet, erfordert ein 224 x 224 Pixel großes Bild mit drei Farbkanälen, um der Anzahl der Eingaben in der ersten Ebene zu entsprechen. Die Anwendung passt jeden Farbwert an, indem sie ihn in eine Zahl zwischen 0 und 1 umwandelt, den Durchschnittswert für diese Farbe subtrahiert und durch die Standardabweichung dividiert. Schließlich kombiniert es die Farbkanäle und wandelt sie in eine NumPy Array, das das Modell verarbeiten kann.

Example application.py— Vorverarbeitung

```
    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)
```

Dieser Prozess liefert die Modellwerte in einem vorhersagbaren Bereich um 0 zentriert. Sie stimmt mit der Vorverarbeitung überein, die auf Bilder im Trainingsdatensatz angewendet wird. Dies ist ein Standardansatz, der jedoch pro Modell variieren kann.

Hochladen von Metriken mit dem SDK für Python

Die Beispielanwendung verwendet das SDK für Python, um Metriken auf Amazon hochzuladen CloudWatchaus.

Example application.py— SDK für 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.")
```

Sie erhält die Berechtigung einer Laufzeitrolle, die Sie während der Bereitstellung zuweisen. Die Rolle wird in der`aws-panorama-sample.yml` AWS CloudFormationVorlage.

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/
```

Die Beispielanwendung installiert das SDK für Python und andere Abhängigkeiten mit pip. Wenn Sie den Anwendungscontainer erstellen, wird der`Dockerfile`führt Befehle aus, um Bibliotheken zusätzlich zu dem, was mit dem Basisimage geliefert wird, zu installieren.

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
```

So verwenden Sie denAWSSDK in Ihrem Anwendungscode ändern Sie zunächst die Vorlage, um Berechtigungen für alle API-Aktionen hinzuzufügen, die die Anwendung verwendet. Aktualisieren desAWS CloudFormationstapeln, indem Sie den`1-create-role.sh`jedes Mal, wenn Sie eine Änderung vornehmen. Stellen Sie dann Änderungen an Ihrem Anwendungscode bereit.

Für Aktionen, die vorhandene Ressourcen ändern oder verwenden, empfiehlt es sich, den Umfang dieser Richtlinie zu minimieren, indem Sie einen Namen oder ein Muster für das Ziel angeben.`Resource`in einer separaten Anweisung. Einzelheiten zu den Aktionen und Ressourcen, die von den einzelnen Services unterstützt werden, finden Sie unterAktionen, Ressourcen und Bedingungsschlüsselin der Service Authorization-Referenz

Nächste Schritte

Anweisungen zur Verwendung der AWS Panorama Application CLI zum Erstellen von Anwendungen und zum Erstellen von Paketen von Grund auf finden Sie in der README der CLI.

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

Weitere Beispielcodes und ein Testdienstprogramm, mit dem Sie Ihren Anwendungscode vor der Bereitstellung validieren können, finden Sie im AWS Panorama Panorama-Beispiel-Repository.

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

Unterstützte Computer Vision-Modelle und -Kameras

AWS Panorama unterstützt ModellePyTorch, die mit Apache MXNet und TensorFlow erstellt wurden. Wenn Sie eine Anwendung bereitstellen, kompiliert AWS Panorama Ihr Modell in SageMaker Neo. Sie können Modelle in Amazon SageMaker oder in Ihrer Entwicklungsumgebung erstellen, sofern Sie Ebenen verwenden, die mit SageMaker Neo kompatibel sind.

Um Videos zu verarbeiten und Bilder zum Senden an ein Modell abzurufen, stellt die AWS Panorama Appliance eine Verbindung zu einem H.264-kodierten Videostream mit dem RTSP-Protokoll her. AWS Panorama testet eine Vielzahl gängiger Kameras auf Kompatibilität.

Abschnitte

Unterstützte Modelle

Wenn Sie eine Anwendung für AWS Panorama erstellen, stellen Sie ein Modell für maschinelles Lernen bereit, das die Anwendung für Computer Vision verwendet. Sie können vorgefertigte und vortrainierte Modelle verwenden, die von Model-Frameworks bereitgestellt werden, ein Beispielmodell oder ein Modell, das Sie selbst erstellen und trainieren.

> **Note**
>
> Eine Liste der vorgefertigten Modelle, die mit AWS Panorama getestet wurden, finden Sie unter Modellkompatibilität.

Wenn Sie eine Anwendung bereitstellen, verwendet AWS Panorama den SageMaker Neo-Compiler, um Ihr Computer Vision-Modell zu kompilieren. SageMakerNeo ist ein Compiler, der Modelle so optimiert, dass sie effizient auf einer Zielplattform ausgeführt werden. Dabei kann es sich um eine Instance in Amazon Elastic Compute Cloud (Amazon EC2) oder ein Edge-Gerät wie die AWS Panorama Appliance handeln.

AWS Panorama unterstützt die Versionen von PyTorch Apache MXNet und TensorFlow die von SageMaker Neo für Edge-Geräte unterstützt werden. Wenn Sie Ihr eigenes Modell erstellen, können Sie die in den SageMakerNeo-Versionshinweisen aufgeführten Framework-Versionen verwenden. In SageMaker können Sie den integrierten Bildklassifizierungsalgorithmus verwenden.

Weitere Informationen zur Verwendung der Modelle in AWS Panorama finden Sie unterComputer Vision-Modelle.

Unterstützte Kameras

Die AWS Panorama Appliance unterstützt H.264-Videostreams von Kameras, die RTSP über ein lokales Netzwerk ausgeben. Bei Kamerastreams mit mehr als 2 Megapixeln verkleinert die Appliance das Bild auf 1920 x 1080 Pixel oder eine gleichwertige Größe, bei der das Seitenverhältnis des Streams erhalten bleibt.

Die folgenden Kameramodelle wurden auf Kompatibilität mit der AWS Panorama Appliance getestet:

- Achse — M3057-PLVE, M3058-PLVE, P1448-LE, P3225-LV Mk II
- LaView— LV-PB3040W
- Vivotek — IB9360-H
- Amcrest — IP2M-841B
- Anoviz — IPC-B850W-S-3X, IPC-D250W-S
- WGCC — PoE Kuppel 4 MP ONVIF

Die Hardwarespezifikationen der Appliance finden Sie unterSpezifikationen zu AWS Panorama Appliances.

Spezifikationen zu AWS Panorama Appliances

Die AWS Panorama Appliance hat die folgenden Hardwarespezifikationen. Für andereKompatible Gerätefinden Sie in der -Dokumentation des Herstellers.

Komponente	Spezifikation
Prozessor und GPU	Nvidia Jetson AGX Xaviermit 32 GB RAM
Ethernet	2x 1000 Base-T (Gigabyte)
USB	1 x USB 2.0 und 1x USB 3.0 Typ A Buchse
HDMI-Ausgang	2,0 a
Dimensionen	197 mm x 243 mm x 40 mm
Gewicht	1,7 lbs (1,7 kg)
Netzteil	100V-240 V 50-60 Hz Wechselstrom 65 W
Eingabewechsel	IEC 60320 C6 (3-polig) Buchse
Staub- und Flüssigkeitsschutz	IP-62
Einhaltung gesetzlicher Vorschriften EMI/EMC	FCC Teil-15 (USA)
Thermische Berührungsgrenzen	IEC-62368
Betriebstemperatur	-20 °C bis 60 °C
Luftfeuchtigkeit bei Betrieb	0% bis 95% RH
Lagertemperatur	-20 °C bis 85 °C
Luftfeuchtigkeit bei Lagerung	Unkontrolliert für niedrige Temperaturen. 90% RH bei hoher Temperatur
Kühlung	Warmluft-Wärmeabsaugung (Lüfter)
Befestigungsoptionen	Rackmount oder freistehend

Komponente	Spezifikation
Netzkabel	1,8 Meter
Leistungssteuerung	Druckknopf
Zurücksetzen	Momentaner Wechsel
Status- und Netzwerk-LEDs	Programmierbare 3-Farben-RGB-LED

Wi-Fi-, Bluetooth- und SD-Kartenspeicher sind auf dem Gerät vorhanden, sind jedoch nicht verwendbar.

Die AWS Panorama Appliance enthält zwei Schrauben zur Montage an einem Server-Rack. Sie können zwei Appliances montieren side-by-side auf einem 19-Zoll-Rack.

Servicekontingente

AWS Panorama wendet Kontingente auf die Ressourcen an, die Sie in Ihrem Konto erstellen, und auf die Anwendungen, die Sie bereitstellen. Wenn Sie AWS Panorama in mehrerenAWSRegionen, Kontingente gelten für jede Region separat. AWS Panorama-Kontingente sind nicht anpassbar.

Zu den Ressourcen in AWS Panorama gehören Geräte, Anwendungsknotenpakete und Anwendungsinstanzen.

- Geräte— Bis zu 50 registrierte Geräte pro Region.
- Knoten-Pakete— 50 Pakete pro Region, mit bis zu 20 Versionen pro Paket.
- Anwendungsinstanzen— Bis zu 10 Anwendungen pro Gerät. Jede Anwendung kann bis zu 8 Kamerastreams überwachen. Die Bereitstellungen sind auf 200 pro Tag für jedes Gerät begrenzt.

Wenn Sie die AWS Panorama Application CLI verwenden,AWS Command Line Interface, oderAWSSDK mit dem AWS Panorama-Service, Kontingente gelten für die Anzahl der API-Aufrufe, die Sie tätigen. Sie können insgesamt bis zu 5 Anfragen pro Sekunde stellen. Für eine Teilmenge von API-Vorgängen, die Ressourcen erstellen oder ändern, gilt ein zusätzliches Limit von 1 Anfrage pro Sekunde.

Eine vollständige Liste der Kontingente finden Sie aufKonsole „Service Quotas“, oder sieheAWS Panorama-Endpunkte und Kontingentein derAllgemeine Amazon Web Services-Referenz.

AWS Panorama-Berechtigungen

Sie können mitAWS Identity and Access Management (IAM) den Zugriff auf denAWS Panorama Service und Ressourcen wie Appliances und Anwendungen steuern. Für Benutzer in Ihrem Konto, die diese verwendenAWS Panorama, verwalten Sie Berechtigungen in einer Berechtigungsrichtlinie, die Sie auf IAM-Rollen anwenden können. Um die Berechtigungen für eine Anwendung zu verwalten, erstellen Sie eine Rolle und weisen sie der Anwendung zu.

Um Berechtigungen für Benutzer in Ihrem Konto zu verwalten, verwenden Sie dieAWS Panorama entsprechende verwaltete Richtlinie oder schreiben Sie Ihre eigene. Sie benötigen Berechtigungen für andereAWS Dienste, um Anwendungs- und Appliance-Protokolle abzurufen, Metriken anzuzeigen und einer Anwendung eine Rolle zuzuweisen.

EineAWS Panorama Appliance verfügt auch über eine Rolle, die ihr die Berechtigung für den Zugriff aufAWS -Services und -Ressourcen gewährt. Die Rolle der Appliance ist eine der Servicerollen, die derAWS Panorama Dienst verwendet, um in Ihrem Namen auf andere Dienste zuzugreifen.

Eine Anwendungsrolle ist eine separate Servicerolle, die Sie für eine Anwendung erstellen, um ihr die Berechtigung zu erteilen,AWS Dienste mit der zu verwendenAWS SDK for Python (Boto). Um eine Anwendungsrolle zu erstellen, benötigen Sie Administratorrechte oder die Hilfe eines Administrators.

Sie können Benutzerberechtigungen nach der Ressource einschränken, auf die sich eine Aktion auswirkt, und in einigen Fällen auch nach zusätzlichen Bedingungen. Sie können z. B. ein Muster für den Amazon-Ressourcennamen (ARN) einer Anwendung festlegen, die von einem Benutzer verlangt, seinen Benutzernamen in den Namen der Anwendungen aufzunehmen, die er erstellt. Informationen zu den Ressourcen und Bedingungen, die von den einzelnen Aktionen unterstützt werden, finden Sie unter Aktionen, Ressourcen undAWS Panorama Konditionsschlüssel für die Serviceberechtigung.

Weitere Informationen finden Sie unter Was ist IAM für? im IAM-Benutzerhandbuch.

Themen

- Identitätsbasierte IAM-Richtlinien für AWS Panorama
- AWS Panorama Panorama-Servicerollen und serviceübergreifende Ressourcen
- Berechtigungen für eine Anwendung erteilen

Identitätsbasierte IAM-Richtlinien für AWS Panorama

Um Benutzern in Ihrem Konto Zugriff auf AWS Panorama zu gewähren, müssen Sie identitätsbasierte Richtlinien inAWS Identity and Access Management (IAM) verwenden. Wenden Sie identitätsbasierte Richtlinien auf IAM-Rollen an, die einem Benutzer zugeordnet sind. Sie können auch Benutzern in einem anderen Konto die Berechtigung erteilen, eine Rolle in Ihrem Konto zu übernehmen und auf Ihre AWS Panorama Panorama-Ressourcen zuzugreifen.

AWS Panorama bietet verwaltete Richtlinien, die Zugriff auf AWS Panorama Panorama-API-Aktionen gewähren und in einigen Fällen Zugriff auf andere -Services, die zur Entwicklung und Verwaltung von AWS Panorama Panorama-Ressourcen verwendet werden. AWS Panorama aktualisiert die verwalteten Richtlinien je nach Bedarf, um sicherzustellen, dass Ihre Benutzer Zugriff auf neue Funktionen haben, sobald diese veröffentlicht werden.

- AWSPanoramaFullAccess— Bietet vollen Zugriff auf AWS Panorama, AWS Panorama Panorama-Zugangspunkte in Amazon S3, Appliance-Anmeldeinformationen inAWS Secrets Manager und Appliance-Logs in Amazon CloudWatch. Beinhaltet die Erlaubnis, eine serviceverknüpfte Rolle für AWS Panorama zu erstellen. Richtlinie ansehen

DieAWSPanoramaFullAccess Richtlinie ermöglicht es Ihnen, AWS Panorama Panorama-Ressourcen zu taggen, verfügt jedoch nicht über alle tagbezogenen Berechtigungen, die von der AWS Panorama Panorama-Konsole verwendet werden. Um -- Berechtigungen zu erteilen, müssen Sie Sie Sie Sie Sie Sie Sie Sie müssen Sie Sie Sie Sie

- ResourceGroupsandTagEditorFullAccess— Richtlinie anzeigen

DieAWSPanoramaFullAccess Richtlinie beinhaltet keine Erlaubnis zum Kauf von Geräten über die AWS Panorama Panorama-Konsole. Um -- Berechtigungen zu erteilen, müssen Sie Sie Sie Sie Sie Sie Sie Sie müssen Sie Sie Sie Sie

- ElementalAppliancesSoftwareFullAccess— Richtlinie anzeigen

Von verwaltete Richtlinien erteilen Berechtigungen für API-Aktionen, ohne die Ressourcen einzuschränken, die ein Benutzer ändern kann. Für eine feinere Steuerung können Sie eigene Richtlinien erstellen, die den Umfang der Berechtigungen eines Benutzers einschränken. Verwenden Sie die Richtlinie für den vollständigen Zugriff als Ausgangspunkt für Ihre Richtlinien.

> **Servicerollen erstellen**
>
> Wenn Sie die AWS Panorama-Konsole zum ersten Mal verwenden, benötigen Sie die Erlaubnis, die von der AWS Panorama Appliance verwendete Servicerolle zu erstellen. Eine Servicerolle erteilt einem Dienst die Berechtigung, Ressourcen zu verwalten oder mit anderen Diensten zu interagieren. Erstellen Sie diese Rolle, bevor Sie Ihren Benutzern Zugriff gewähren.

Einzelheiten zu den Ressourcen und Bedingungen, mit denen Sie den Umfang der Benutzerberechtigungen in AWS Panorama einschränken können, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Panorama in der Service Authorization Reference.

AWS Panorama Panorama-Servicerollen und serviceübergreifende Ressourcen

AWS Panorama verwendet andere AWS-Services, um die AWS Panorama Appliance zu verwalten, Daten zu speichern und Anwendungsressourcen zu importieren. Eine Servicerolle erteilt einem Dienst die Berechtigung, Ressourcen zu verwalten oder mit anderen Diensten zu interagieren. Wenn Sie sich erstmals bei der AWS Panorama Panorama-Konsole anmelden, erstellen Sie die folgenden Servicerollen:

- AWSServiceRoleForAWSPanorama— Ermöglicht AWS Panorama die Verwaltung von Ressourcen in AWS IoT, AWS Secrets Manager und AWS Panorama.

  Aktualisierung der verwalteten Richtlinie:AWSPanoramaServiceLinkedRolePolicy
- AWSPanoramaApplianceServiceRole— Ermöglicht einer AWS Panorama Appliance das Hochladen von Protokollen auf CloudWatchund um Objekte von Amazon S3 S3-Zugriffspunkten abzurufen, die von AWS Panorama erstellt wurden.

  Aktualisierung der verwalteten Richtlinie:AWSPanoramaApplianceServiceRolePolicy

Um die Berechtigungen anzuzeigen, die jeder Rolle zugewiesen sind, verwenden Sie dieIAM-Konsole. Wo immer möglich, sind die Berechtigungen der Rolle auf Ressourcen beschränkt, die einem von AWS Panorama verwendeten Benennungsmuster entsprechen. Beispiel,`AWSServiceRoleForAWSPanorama`gewährt nur Zugriffsberechtigung für den DienstAWS IoTRessourcen, die haben`panorama`in ihrem Namen.

Abschnitte

- Absichern der Appliance-Rolle
- Nutzung anderer Services

Absichern der Appliance-Rolle

Die AWS Panorama Appliance verwendet die`AWSPanoramaApplianceServiceRole`Rolle für den Zugriff auf Ressourcen in Ihrem Konto. Die Appliance hat die Berechtigung zum Hochladen der Protokolle in CloudWatch Protokolliert, liest Kamerastream-AnmeldeinformationenAWS Secrets Managerund um auf Anwendungsartefakte in Amazon Simple Storage Service (Amazon S3) - Zugriffspunkten zuzugreifen, die AWS Panorama erstellt.

> **Note**
>
> Anwendungen verwenden nicht die Berechtigungen der Appliance. So geben Sie Ihrer Anwendung die Erlaubnis zur VerwendungAWSDienste, erstellen Sie eineRolle der Anwendung.

AWS Panorama verwendet dieselbe Servicerolle für alle Appliances in Ihrem Konto und verwendet keine Rollen kontoübergreifend. Für eine zusätzliche Sicherheitsebene können Sie die Vertrauensrichtlinie der Appliance-Rolle ändern, um dies explizit durchzusetzen. Dies ist eine bewährte Methode, wenn Sie Rollen verwenden, um einem Dienst die Berechtigung für den Zugriff auf Ressourcen in Ihrem Konto zu erteilen.

So aktualisieren Sie die Vertrauensrichtlinie für Appliance-Rollen

1. Öffnen Sie die Appliance-Rolle in der IAM-Konsole:AWSPanoramaApplianceServiceRole
2. Wählen Sie Edit Trust Relationship (Vertrauensstellungen bearbeiten).
3. Aktualisieren Sie den Inhalt der Richtlinie und wählen Sie dannAktualisieren der Vertrauensrichtlinie.

Die folgende Vertrauensrichtlinie enthält eine Bedingung, die sicherstellt, dass AWS Panorama die Appliance-Rolle für eine Appliance in Ihrem Konto übernimmt. Die`aws:SourceAccount`condition vergleicht die von AWS Panorama angegebene Konto-ID mit der Konto-ID, die Sie in die Richtlinie aufnehmen.

Example Vertrauensrichtlinie — Spezifisches Konto

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

```
        }
      }
    }
  ]
}
```

Wenn Sie AWS Panorama weiter einschränken und zulassen möchten, dass AWS Panorama nur die Rolle mit einem bestimmten Gerät übernimmt, können Sie das Gerät anhand des ARN angeben. Die`aws:SourceArn`condition vergleicht den ARN der von AWS Panorama angegebenen Appliance mit dem ARN, den Sie in die Richtlinie aufnehmen.

Example Vertrauensrichtlinie — Einzelgerät

```
{
  "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"
        }
      }
    }
  ]
}
```

Wenn Sie die Appliance zurücksetzen und erneut bereitstellen, müssen Sie die Quell-ARN-Bedingung vorübergehend entfernen und sie dann erneut mit der neuen Geräte-ID hinzufügen.

Weitere Informationen zu diesen Bedingungen und bewährten Sicherheitsmethoden beim Verwenden von Rollen zum Zugriff auf Ressourcen in Ihrem Konto durch Services finden Sie unterDas Problem des verwirrten Stellvertretersim IAM-Benutzerhandbuch.

Nutzung anderer Services

AWS Panorama erstellt Ressourcen in den folgenden Services oder greift darauf zu:

- AWS IoT— Dinge, Richtlinien, Zertifikate und Aufgaben für die AWS Panorama Appliance
- Amazon S3— Zugriffspunkte für die Bereitstellung von Anwendungsmodellen, Code und Konfigurationen.
- Secrets Manager— Kurzfristige Anmeldeinformationen für die AWS Panorama Appliance.

Informationen zum Amazon-Ressourcenname (ARN) -Format oder zu den Berechtigungsbereichen für jeden Service finden Sie in den Themen imIAM User Guideauf die in dieser Liste verlinkt ist.

Berechtigungen für eine Anwendung erteilen

Sie können eine Rolle für Ihre Anwendung erstellen, um ihr die Berechtigung zum Aufrufen zu erteilenAWS-Services. Standardmäßig haben Anwendungen keine anderen Berechtigungen. Sie erstellen eine Anwendungsrolle in IAM und weisen sie während der Bereitstellung einer Anwendung zu. Um Ihrer Anwendung nur die Berechtigungen zu gewähren, die sie benötigt, erstellen Sie eine Rolle mit Berechtigungen für bestimmte API-Aktionen.

Die-Beispielanwendungenthält einAWS CloudFormationVorlage und Skript, die eine Anwendungsrolle erstellen. Es ist ein-Servicerollevon dem AWS Panorama annehmen kann. Diese Rolle erteilt der Anwendung die Berechtigung, CloudWatch aufzurufen, um Metriken hochzuladen.

Example aws-panorama-beispiel.yml— Rolle für die Anwendung

```
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/
```

Sie können dieses Skript erweitern, um Berechtigungen für andere Dienste zu erteilen, indem Sie eine Liste von API-Aktionen oder -Mustern für den Wert von`Action`aus.

Weitere Informationen zu Berechtigungen in AWS Panorama finden Sie unterAWS Panorama-Berechtigungenaus.

Verwalten derAWS PanoramaAppliances

DieAWS PanoramaAppliance ist die Hardware, die Ihre Anwendungen ausführt. Sie benutzen dasAWS Panorama-Konsole, um eine Appliance zu registrieren, ihre Software zu aktualisieren und Anwendungen darauf bereitzustellen. Die Software auf demAWS PanoramaDie Appliance stellt eine Verbindung zu Kamera-Streams her, sendet Videobilder an Ihre Anwendung und zeigt die Videoausgabe auf einem angeschlossenen Display an.

Nach dem Einrichten Ihrer oder anderen Appliancekompatibles Gerätregistrieren, registrieren Sie Kameras für die Verwendung mit Anwendungen. SieVerwalten Sie Kamera-StreamsimAWS Panoramaconsole. Wenn Sie eine Anwendung bereitstellen, wählen Sie aus, welche Kamera-Streams die Appliance zur Verarbeitung an sie sendet.

Für Tutorials, die dasAWS PanoramaAppliance mit einer Beispielanwendung, sieheErste Schritte mit AWS Panoramaaus.

Themen

Verwaltung einer AWS Panorama Panorama-Appliance

Sie verwenden die AWS Panorama-Konsole, um die AWS Panorama Appliance und andere kompatible Geräte zu konfigurieren, zu aktualisieren oder zu deregistrieren.

Folgen Sie den Anweisungen im Tutorial „Erste Schritte“, um eine Appliance einzurichten. Beim Einrichtungsprozess werden die Ressourcen in AWS Panorama erstellt, die Ihre Appliance verfolgen und Updates und Bereitstellungen koordinieren.

Informationen zur Registrierung einer Appliance bei der AWS Panorama Panorama-API finden Sie unterAutomatisieren Sie die Geräteregistrierung.

Abschnitte

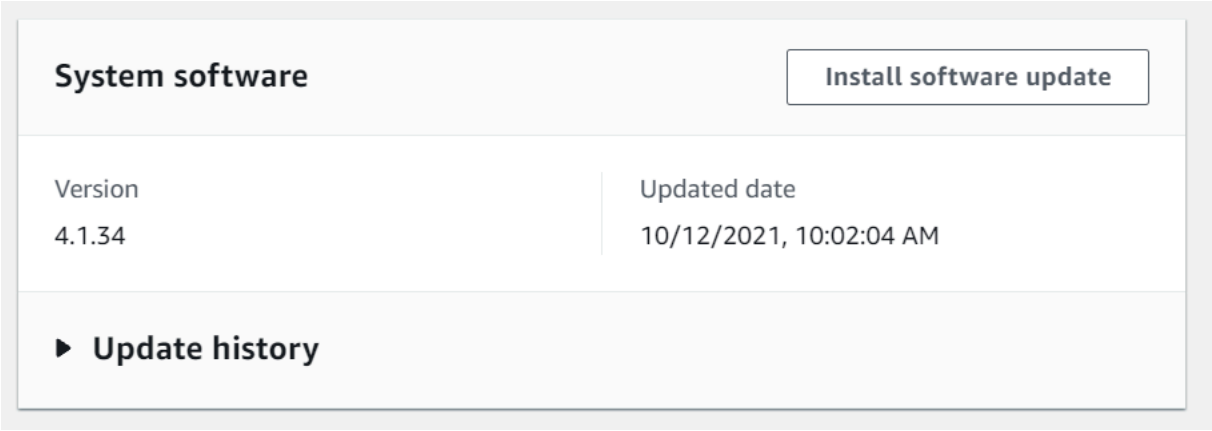
- Aktualisieren Sie die Appliance-Software
- Einen Appliance abmelden
- Neustart einer Appliance
- Zurücksetzen einer Appliance

Aktualisieren Sie die Appliance-Software

Sie können Softwareupdates für die Appliance in der AWS Panorama Panorama-Konsole anzeigen und bereitstellen. Updates können erforderlich oder optional sein. Wenn ein erforderliches Update verfügbar ist, werden Sie von der Konsole aufgefordert, es zu installieren. Sie können optionale Updates auf der Seite mit den Appliance-Einstellungen anwenden.

Um die Appliance-Software zu aktualisieren

1. Öffnen Sie die Seite „Geräte“ der AWS Panorama Panorama-Konsole.
2. Wählen Sie ein Gerät.
3. Wählen Sie Einstellungen
4. Wählen Sie unter Systemsoftware die Option Softwareupdate installieren aus.



5. Wählen Sie eine neue Version und dann Installieren.

Einen Appliance abmelden

Wenn Sie mit der Arbeit mit einer Appliance fertig sind, können Sie die AWS Panorama Panorama-Konsole verwenden, um sie abzumelden und die zugehörigenAWS IoT Ressourcen zu löschen.

Um eine Appliance zu löschen

1. Öffnen Sie die Seite „Geräte“ der AWS Panorama Panorama-Konsole.
2. Wählen Sie den Namen der Appliance.
3. Wählen Sie Löschen.
4. Geben Sie den Namen der Appliance ein und wählen Sie Löschen.

Wenn Sie eine Appliance aus dem AWS Panorama Panorama-Service löschen, werden Daten auf der Appliance nicht automatisch gelöscht. Eine abgemeldete Appliance kann keine Verbindung zuAWS Diensten herstellen und kann erst erneut registriert werden, wenn sie zurückgesetzt wurde.

Neustart einer Appliance

Sie können eine Appliance remote neu starten.

So starten Sie eine Appliance neu:

1. Öffnen Sie die Seite „Geräte“ der AWS Panorama Panorama-Konsole.
2. Wählen Sie den Namen der Appliance.
3. Wählen Sie Reboot.

Die Konsole sendet eine Nachricht an die Appliance, um sie neu zu starten. Zum Empfang des Signals muss das Gerät eine Verbindung herstellen könnenAWS IoT. Informationen zum Neustart einer Appliance mit der AWS Panorama Panorama-API finden Sie unterAppliances neu starten.

Zurücksetzen einer Appliance

Wenn eine Appliance in einer anderen Region oder mit einem anderen Konto verwendet werden soll, müssen Sie sie zurücksetzen und erneut ein neues Zertifikat bereitstellen. Beim Zurücksetzen des Geräts wird die neueste erforderliche Softwareversion angewendet und alle Kontodaten werden gelöscht.

Um einen Reset-Vorgang zu starten, muss das Gerät angeschlossen und ausgeschaltet sein. Halten Sie sowohl die Ein-/Aus-Taste als auch die Reset-Taste fünf Sekunden lang gedrückt. Wenn Sie die Tasten loslassen, blinkt die Statusanzeige orange. Warten Sie, bis die Statusanzeige grün blinkt, bevor Sie die Appliance bereitstellen oder die Verbindung trennen.

Sie können die Appliance-Software auch zurücksetzen, ohne Zertifikate vom Gerät zu löschen. Weitere Informationen finden Sie unter Ein-/Ausschalttasten.

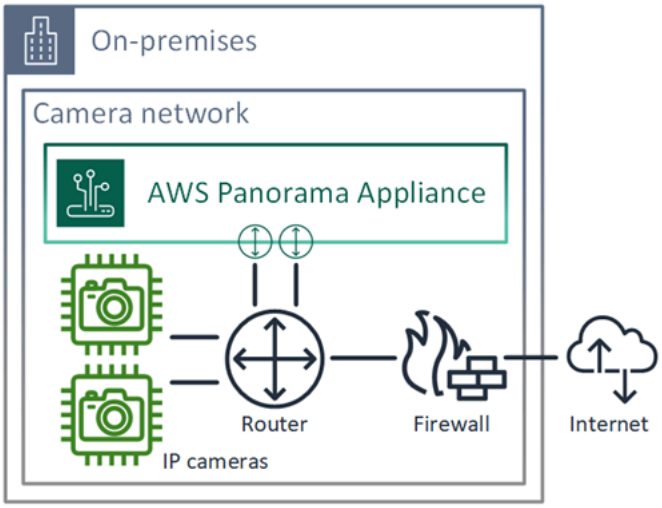
Die AWS Panorama Appliance mit Ihrem Netzwerk verbinden

Die AWS Panorama Appliance erfordert Konnektivität sowohl mit der AWS Cloud als auch mit Ihrem lokalen Netzwerk von IP-Kameras. Sie können die Appliance mit einer einzigen Firewall verbinden, die Zugriff auf beide gewährt, oder jede der beiden Netzwerkschnittstellen des Geräts mit einem anderen Subnetz verbinden. In beiden Fällen müssen Sie die Netzwerkverbindungen der Appliance sichern, um unbefugten Zugriff auf Ihre Kamerastreams zu verhindern.

Sections

Eine einzige Netzwerkkonfiguration

Die Appliance verfügt über zwei Ethernet-Ports. Wenn Sie den gesamten Datenverkehr zum und vom Gerät über einen einzigen Router weiterleiten, können Sie den zweiten Port zur Redundanz verwenden, falls die physische Verbindung zum ersten Port unterbrochen wird. Konfigurieren Sie Ihren Router so, dass die Appliance nur eine Verbindung zu Kamerastreams und dem Internet herstellen kann und dass Kamerastreams Ihr internes Netzwerk nicht verlassen.

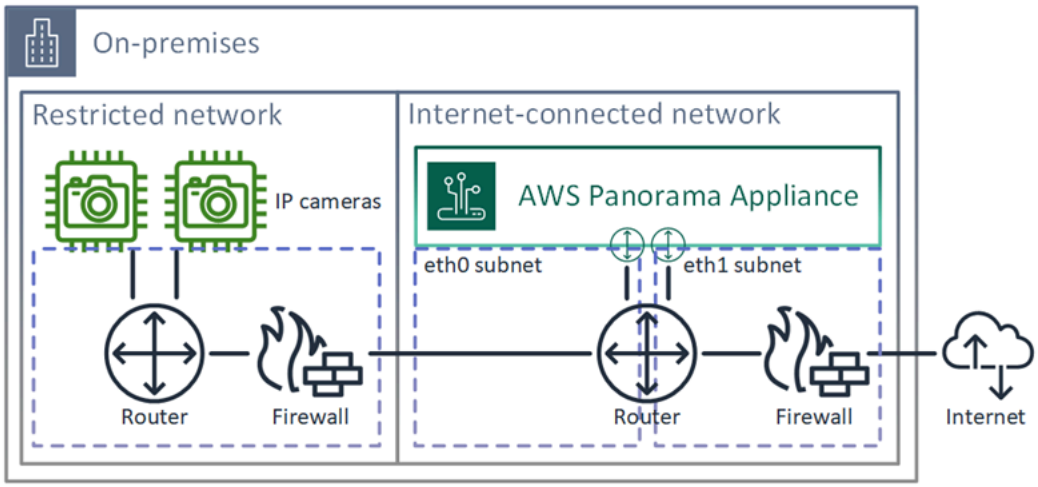


Einzelheiten zu den Ports und Endpunkten, auf die die Appliance Zugriff benötigt, finden Sie unter Konfiguration des Servicezugriffs undKonfiguration des lokalen Netzwerkzugriffs.

Duale Netzwerkkonfiguration

Für eine zusätzliche Sicherheitsebene können Sie das Gerät getrennt von Ihrem Kameranetzwerk in einem mit dem Internet verbundenen Netzwerk platzieren. Eine Firewall zwischen Ihrem eingeschränkten Kameranetzwerk und dem Netzwerk der Appliance ermöglicht der Appliance nur den Zugriff auf Videostreams. Wenn Ihr Kameranetzwerk zuvor aus Sicherheitsgründen über Air-Gaps verfügte, ziehen Sie diese Methode möglicherweise der Verbindung des Kameranetzwerks mit einem Router vor, der auch Zugriff auf das Internet gewährt.

Das folgende Beispiel zeigt, wie das Gerät an jedem Port eine Verbindung zu einem anderen Subnetz herstellt. Der Router platziert die `eth0` Schnittstelle in einem Subnetz, das zum Kameranetzwerk weiterleitet, und `eth1` in einem Subnetz, das zum Internet weiterleitet.



Sie können die IP-Adresse und MAC-Adresse jedes Ports in der AWS-Panorama-Konsole bestätigen.

Konfiguration des Servicezugriffs

Während der Bereitstellung können Sie die Appliance so konfigurieren, dass sie eine bestimmte IP-Adresse anfordert. Wählen Sie im Voraus eine IP-Adresse aus, um die Firewall-Konfiguration zu vereinfachen und sicherzustellen, dass sich die Adresse der Appliance nicht ändert, wenn sie für einen längeren Zeitraum offline ist.

Die Appliance verwendet AWS Dienste, um Softwareupdates und -bereitstellungen zu koordinieren. Konfigurieren Sie Ihre Firewall so, dass die Appliance eine Verbindung zu diesen Endpunkten herstellen kann.

Internetzugang

- AWS IoT(HTTPS und MQTT, Ports 443, 8443 und 8883) — AWS IoT Core und Endpunkte für die Geräteverwaltung. Einzelheiten finden Sie unter Endpunkte und Kontingente für AWS IoT Device Management in derAllgemeine Amazon Web Services-Referenz.
- AWS IoTAnmeldeinformationen (HTTPS, Port 443) — `credentials.iot.<region>.amazonaws.com` und Subdomains.
- Amazon Elastic Container Registry (HTTPS, Port 443) — `api.ecr.<region>.amazonaws.com` `dkr.ecr.<region>.amazonaws.com` und Subdomains.
- 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` und Subdomains.

Wenn Ihre Anwendung andere AWS Dienste aufruft, benötigt die Appliance ebenfalls Zugriff auf die Endpunkte für diese Dienste. Weitere Informationen finden Sie unter Dienstendpunkte und Kontingente.

Konfiguration des lokalen Netzwerkzugriffs

Die Appliance benötigt lokalen Zugriff auf RTSP-Videostreams, jedoch nicht über das Internet. Konfigurieren Sie Ihre Firewall so, dass die Appliance intern auf RTSP-Streams an Port 554 zugreifen kann und dass keine Streams in das Internet ein- oder ausgehen können.

Lokaler Zugriff

- Echtzeit-Streaming-Protokoll (RTSP, Port 554) — Zum Lesen von Kamerastreams.
- Network Time Protocol (NTP, Port 123) — Um die Uhr der Appliance synchron zu halten. Wenn Sie in Ihrem Netzwerk keinen NTP-Server betreiben, kann die Appliance auch über das Internet eine Verbindung zu öffentlichen NTP-Servern herstellen.

Private Konnektivität

Die AWS Panorama Appliance benötigt keinen Internetzugang, wenn Sie sie in einem privaten VPC-Subnetz mit einer VPN-Verbindung bereitstellen. AWS Sie können Site-to-Site VPN verwenden oder AWS Direct Connect um eine VPN-Verbindung zwischen einem lokalen Router und herzustellen.

AWS In Ihrem privaten VPC-Subnetz erstellen Sie Endpoints, über die sich die Appliance mit Amazon Simple Storage Service und anderen Services verbinden kann. AWS IoT Weitere Informationen finden Sie unter Eine Appliance mit einem privaten Subnetz verbinden.

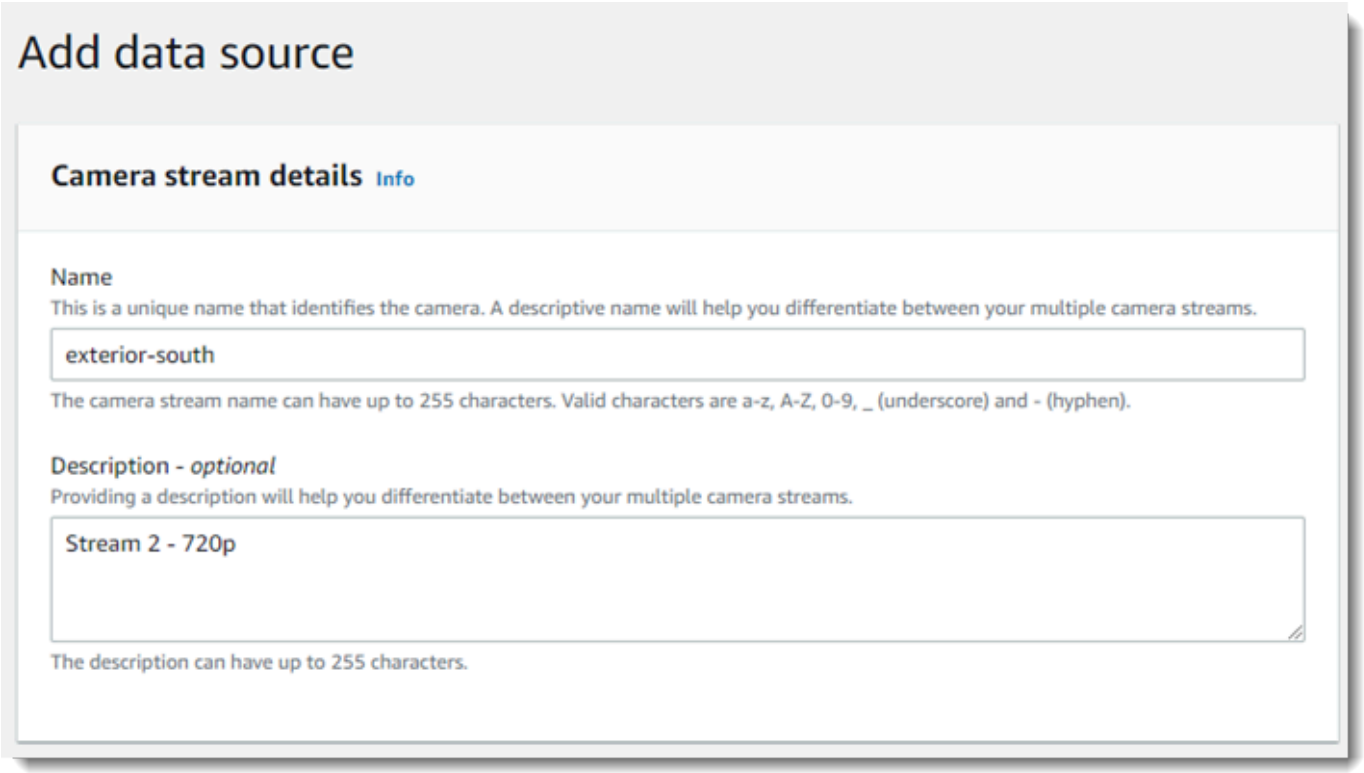
Verwalten von Kamera-Streams in AWS Panorama

Verwenden Sie die AWS Panorama-Konsole, um Videostreams als Datenquellen für Ihre Anwendung zu registrieren. Eine Anwendung kann mehrere Streams gleichzeitig verarbeiten und mehrere Appliances können sich mit demselben Stream verbinden.

> **Important**
>
> Eine Anwendung kann sich mit jedem Kamerastream verbinden, der über das lokale Netzwerk routinierbar ist, mit dem sie verbunden ist. Um Ihre Videostreams zu sichern, konfigurieren Sie Ihr Netzwerk so, dass nur RTSP-Datenverkehr lokal zugelassen wird. Weitere Informationen finden Sie unter Sicherheit in AWS Panorama.

So registrieren Sie einen Kamera-Stream

1. Öffnen Sie die AWS Panorama Panorama-KonsoleSeite „Datenquellen“aus.
2. Klicken Sie aufDatenquelle hinzufügenaus.



3. Konfigurieren Sie die folgenden Einstellungen.

- Name— Ein Name für den Kamera-Stream.
- Beschreibung— Eine kurze Beschreibung der Kamera, ihres Standorts oder anderer Details.
- RTSP-URL— Eine URL, die die IP-Adresse der Kamera und den Pfad zum Stream angibt. Beispiel: `rtsp://192.168.0.77/live/mpeg4/`
- Erweitern Sie im angezeigten Detailbereich die Option— Falls der Kamera-Stream passwortgeschützt ist, geben Sie den Benutzernamen und das Kennwort an.

4. Wählen Sie Save (Speichern) aus.

Informationen zum Registrieren eines Kamera-Streams bei der AWS Panorama API finden Sie unterAutomatisieren Sie die Geräteregistrierungaus.

Eine Liste der Kameras, die mit der AWS Panorama Appliance kompatibel sind, finden Sie unterUnterstützte Computer Vision-Modelle und -Kameraslaus.

Einen Stream entfernen

Sie können einen Kamera-Stream in der AWS Panorama Panorama-Konsole löschen.

So entfernen Sie einen Kamera-Stream

1. Öffnen Sie die AWS Panorama Panorama-KonsoleSeite „Datenquellen“aus.
2. Wähle einen Kamera-Stream.
3. Klicken Sie aufDatenquelle löschenaus.

Das Entfernen eines Kamerastreams aus dem Dienst stoppt nicht die Ausführung von Anwendungen oder löscht Kamera-Anmeldeinformationen aus Secrets Manager nicht. Um Secrets zu löschen, verwenden Sie dieSecrets-Manager-Konsoleaus.

Verwalten Sie Anwendungen auf einer AWS Panorama Panorama-Appliance

Eine Anwendung ist eine Kombination aus Code, Modellen und Konfiguration. Von der-Gerätein der AWS Panorama Panorama-Konsole können Sie Anwendungen auf der Appliance verwalten.

So verwalten Sie Anwendungen auf einer AWS Panorama Panorama-Appliance

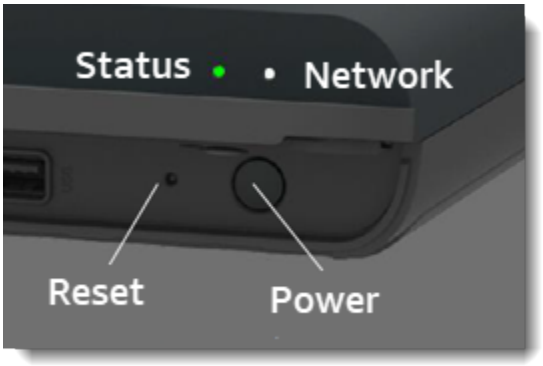
1. Öffnen Sie die AWS Panorama Panorama-KonsoleSeite „Geräte“aus.
2. Wählen Sie ein Gerät aus.

DieBereitgestellte Anwendungenzeigt Anwendungen an, die auf der Appliance bereitgestellt wurden.

Verwenden Sie die Optionen auf dieser Seite, um bereitgestellte Anwendungen von der Appliance zu entfernen oder eine ausgeführte Anwendung durch eine neue Version zu ersetzen. Sie können eine Anwendung auch klonen (ausgeführt oder gelöscht), um eine neue Kopie davon bereitzustellen.

Tasten und Lichter der AWS Panorama Appliance

Die AWS Panorama Appliance verfügt über zwei LED-Leuchten über dem Netzschalter, die den Gerätestatus und die Netzwerkkonnektivität anzeigen.



Statusanzeige

Die LEDs ändern ihre Farbe und blinken, um den Status anzuzeigen. Ein langsames Blinken erfolgt einmal alle drei Sekunden. Ein schnelles Blinken erfolgt einmal pro Sekunde.

Status-LED-Status

- Blinkt schnell Grün— Die Appliance wird gestartet.
- Festes Grün— Das Gerät funktioniert normal.
- Blow-Blau blinkend— Die Appliance kopiert Konfigurationsdateien und versucht, sich bei zu registrierenAWS IoT.
- Blinkt schnell Blau— Das Gerät isteinen Logbild kopierenauf ein USB-Laufwerk.
- Blinkt schnell Rot— Das Gerät ist beim Start auf einen Fehler gestoßen oder es ist überhitzt.
- Slow-orange blinkend— Die Appliance stellt die neueste Softwareversion wieder her.
- Blinkt schnell orange— Die Appliance stellt die Mindestversion der Software wieder her.

Netzwerkbeleuchtung

Die Netzwerk-LED hat die folgenden Status:

Netzwerkzustände

- Festes Grün— Ein Ethernet-Kabel ist angeschlossen.

- Grün blinkend— Die Appliance kommuniziert über das Netzwerk.
- Festes Rot— Ein Ethernet-Kabel ist nicht angeschlossen.

Ein-/Ausschalttasten

Die Power- und Reset-Tasten befinden sich an der Vorderseite des Geräts unter einer Schutzabdeckung. Der Reset-Knopf ist kleiner und versenkt. Drücken Sie mit einem kleinen Schraubenzieher oder einer Büroklammer darauf.

So setzen Sie ein Gerät zurück

1. Das Gerät muss angeschlossen und ausgeschaltet sein. Um das Gerät auszuschalten, halten Sie den Betriebsschalter 1 Sekunde lang gedrückt und warten Sie, bis die Abschaltsequenz abgeschlossen ist. Die Abschaltsequenz dauert etwa 10 Sekunden.
2. Verwenden Sie die folgenden Tastenkombinationen, um die Appliance zurückzusetzen. Ein kurzer Druck dauert 1 Sekunde. Ein langes Drücken beträgt 5 Sekunden. Bei Operationen, die mehrere Tasten erfordern, halten Sie beide Tasten gleichzeitig gedrückt.

   - Vollständiger Reset— Drücken Sie lange auf Power und Reset.

     Stellt die Mindestversion der Software wieder her und löscht alle Konfigurationsdateien und Anwendungen.
   - Neuste Softwareversion wiederherstellen— Drücken Sie kurz auf Reset.

     Wendet das neueste Softwareupdate erneut auf die Appliance an.
   - Wiederherstellung der Mindestversion der Software— Drücken Sie lange auf Reset.

     Wendet das neueste erforderliche Softwareupdate erneut auf die Appliance an.
3. Lassen Sie beide Tasten los. Das Gerät schaltet sich ein und die Statusanzeige blinkt mehrere Minuten lang orange.
4. Wenn das Gerät bereit ist, blinkt die Statusanzeige grün.

Durch das Zurücksetzen einer Appliance wird sie nicht aus dem AWS Panorama Panorama-Service gelöscht. Weitere Informationen finden Sie unter Einen Appliance abmelden.

Verwalten vonAWS PanoramaAnwendungen

Anwendungen laufen auf demAWS PanoramaGerät zur Ausführung von Computer-Vision-Aufgaben in Videostreams. Sie können Computer-Vision-Anwendungen erstellen, indem Sie Python-Code und Modelle für maschinelles Lernen kombinieren und sie auf demAWS PanoramaGerät über das Internet. Anwendungen können Videos an eine Anzeige senden oder das AWS SDK verwenden, um Ergebnisse an AWS-Services zu senden.

Themen

Eine Anwendung bereitstellen

Um eine Anwendung bereitzustellen, verwenden Sie die AWS Panorama Application CLI, importieren sie in Ihr Konto, erstellen den Container, laden Assets hoch und registrieren sie und erstellen eine Anwendungsinstanz. Dieses Thema geht detailliert auf jeden dieser Schritte ein und beschreibt, was im Hintergrund vor sich geht.

Wenn Sie noch keine Anwendung bereitgestellt haben, finden Sie Erste Schritte mit AWS Panorama eine exemplarische Vorgehensweise.

Weitere Informationen zum Anpassen und Erweitern der Beispielanwendung finden Sie unterErstellungAWS PanoramaAnwendungen.

Abschnitte

- Installieren Sie die AWS Panorama-Anwendungs-CLI
- Eine Anwendung importieren
- Erstellen Sie ein Container-Image
- Ein Modell importieren
- Laden Sie Anwendungsressourcen hoch
- Stellen Sie eine Anwendung mit der AWS Panorama-Konsole bereit
- Automatisieren Sie die Anwendungsbereitstellung

Installieren Sie die AWS Panorama-Anwendungs-CLI

Um die AWS Panorama Application CLI zu installierenAWS CLI, verwenden Sie pip.

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

Um Anwendungs-Images mit der AWS Panorama Application CLI zu erstellen, benötigen Sie Docker. Unter Linux sind `qemu` auch verwandte Systembibliotheken erforderlich. Weitere Informationen zur Installation und Konfiguration der AWS Panorama Application CLI finden Sie in der README-Datei im Projekt-Repository. GitHub

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

Anweisungen zum Einrichten einer Build-Umgebung in Windows mit WSL2 finden Sie unter. Einrichten einer Entwicklungsumgebung in Windows

Eine Anwendung importieren

Wenn Sie mit einer Beispielanwendung oder einer von einem Drittanbieter bereitgestellten Anwendung arbeiten, verwenden Sie die AWS Panorama Application CLI, um die Anwendung zu importieren.

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

Dieser Befehl benennt Anwendungspakete mit Ihrer Konto-ID um. Paketnamen beginnen mit der Konto-ID des Kontos, für das sie bereitgestellt werden. Wenn Sie eine Anwendung für mehrere Konten bereitstellen, müssen Sie die Anwendung für jedes Konto separat importieren und verpacken.

Zum Beispiel die Beispielanwendung dieses Handbuchs, ein Codepaket und ein Modellpaket, die jeweils mit einer Platzhalter-Konto-ID benannt sind. Der `import-application` Befehl benennt diese um, sodass sie die Konto-ID verwenden, die die CLI aus den Anmeldeinformationen Ihres Workspace ableitet. 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`wird in den Paketverzeichnisnamen und im Anwendungsmanifest (`graph.json`), das sich auf sie bezieht, durch Ihre Konto-ID ersetzt. Sie können Ihre Konto-ID bestätigen, indem Sie `aws sts get-caller-identity` mit dem anrufenAWS CLI.

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

Erstellen Sie ein Container-Image

Ihr Anwendungscode ist in einem Docker-Container-Image verpackt, das den Anwendungscode und die Bibliotheken enthält, die Sie in Ihrem Dockerfile installieren. Verwenden Sie den `build-container` CLI-Befehl AWS Panorama Application, um ein Docker-Image zu erstellen und ein Dateisystem-Image zu exportieren.

```
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
```

Dieser Befehl erstellt ein Docker-Image mit dem Namen `code_asset` und exportiert ein Dateisystem in ein `.tar.gz` Archiv im `assets` Ordner. Die CLI ruft das Basisimage der Anwendung aus der Amazon Elastic Container Registry (Amazon ECR) ab, wie in der Dockerfile der Anwendung angegeben.

Zusätzlich zum Container-Archiv erstellt die CLI ein Asset für den Paketdeskriptor ()`descriptor.json`. Beide Dateien werden mit einer eindeutigen Kennung umbenannt, die einen Hash der Originaldatei widerspiegelt. Die AWS Panorama Application CLI fügt der Paketkonfiguration

außerdem einen Block hinzu, der die Namen der beiden Assets aufzeichnet. Diese Namen werden von der Appliance während des Bereitstellungsprozesses verwendet.

Example packages/123456789012-sample_code-1.0/package.json — mit Asset-Block

```
{
    "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"
                    },
```

Der im `build-container` Befehl angegebene Name des Code-Assets muss mit dem Wert des `asset` Felds in der Paketkonfiguration übereinstimmen. Im vorherigen Beispiel sind beide Werte`code_asset`.

Ein Modell importieren

Ihre Anwendung hat möglicherweise ein Modellarchiv in ihrem Assets-Ordner oder das Sie separat herunterladen. Wenn Sie ein neues Modell, ein aktualisiertes Modell oder eine aktualisierte Modelldeskriptordatei haben, verwenden Sie den `add-raw-model` Befehl, um es zu importieren.

```
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
```

Wenn Sie nur die Deskriptordatei aktualisieren müssen, können Sie das vorhandene Modell im Assets-Verzeichnis wiederverwenden. Möglicherweise müssen Sie die Deskriptordatei aktualisieren, um Funktionen wie den Fließkommagenauigkeitsmodus zu konfigurieren. Das folgende Skript zeigt beispielsweise, wie Sie dies mit der Beispiel-App tun.

Example util-Scripts/ .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
```

Änderungen an der Deskriptordatei im Modellpaketverzeichnis werden erst angewendet, wenn Sie sie mit der CLI erneut importieren. Die CLI aktualisiert die Modellpaketkonfiguration mit den neuen Asset-Namen direkt, ähnlich wie sie die Konfiguration für das Anwendungscodepaket aktualisiert, wenn Sie einen Container neu erstellen.

Laden Sie Anwendungsressourcen hoch

Verwenden Sie den Befehl, um die Assets der Anwendung hochzuladen und zu registrieren, zu denen das Modellarchiv, das Container-Dateisystemarchiv und die `package-application` zugehörigen Deskriptordateien gehören.

```
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
...
```

Wenn an einer Asset-Datei oder der Paketkonfiguration keine Änderungen vorgenommen werden, überspringt die CLI dies.

```
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
```

Die CLI lädt die Assets für jedes Paket auf einen Amazon S3-Zugangspunkt hoch, der für Ihr Konto spezifisch ist. AWS Panorama verwaltet den Access Point für Sie und stellt über die DescribePackageAPI Informationen darüber bereit. Die CLI lädt die Assets für jedes Paket an den für das Paket bereitgestellten Speicherort hoch und registriert sie mit den in der Paketkonfiguration beschriebenen Einstellungen beim AWS Panorama-Service.

Stellen Sie eine Anwendung mit der AWS Panorama-Konsole bereit

Sie können eine Anwendung mit der AWS Panorama-Konsole bereitstellen. Während des Bereitstellungsprozesses wählen Sie aus, welche Kamerastreams an den Anwendungscode übergeben werden sollen, und konfigurieren die vom Entwickler der Anwendung bereitgestellten Optionen.

Um eine Anwendung bereitzustellen

1. Öffnen Sie die Seite „Bereitgestellte Anwendungen“ der AWS Panorama-Konsole.
2. Wählen Sie Anwendung bereitstellen aus.
3. Fügen Sie den Inhalt des Anwendungsmanifests`graph.json`, in den Texteditor ein. Wählen Sie Weiter.
4. Geben Sie einen Namen und eine Beschreibung ein.
5. Wählen Sie Proceed to deploy aus.
6. Wählen Sie Bereitstellung starten aus.
7. Wenn Ihre Anwendung eine Rolle verwendet, wählen Sie sie aus dem Drop-down-Menü aus. Wählen Sie Weiter.
8. Wählen Sie Gerät auswählen und dann Ihr Gerät aus. Wählen Sie Weiter.
9. Wählen Sie im Schritt Datenquellen auswählen die Option Eingabe (en) anzeigen aus und fügen Sie Ihren Kamerastream als Datenquelle hinzu. Wählen Sie Weiter.
10. Konfigurieren Sie im Schritt Konfigurieren alle vom Entwickler definierten anwendungsspezifischen Einstellungen. Wählen Sie Weiter.
11. Wählen Sie Deploy und dann Fertig.
12. Wählen Sie in der Liste der bereitgestellten Anwendungen die Anwendung aus, deren Status überwacht werden soll.

Der Bereitstellungsprozess dauert 15 bis 20 Minuten. Die Ausgabe der Appliance kann für einen längeren Zeitraum leer sein, während die Anwendung gestartet wird. Wenn Sie auf einen Fehler stoßen, finden Sie weitere Informationen unterFehlerbehebung.

Automatisieren Sie die Anwendungsbereitstellung

Sie können den Prozess der Anwendungsbereitstellung mit der CreateApplicationInstanceAPI automatisieren. Die API verwendet zwei Konfigurationsdateien als Eingabe. Das Anwendungsmanifest spezifiziert die verwendeten Pakete und ihre Beziehungen. Die zweite Datei ist eine Überschreibungsdatei, die die Außerkraftsetzung von Werten im Anwendungsmanifest während der Bereitstellung festlegt. Mithilfe einer Overrides-Datei können Sie dasselbe Anwendungsmanifest verwenden, um die Anwendung mit verschiedenen Kamerastreams bereitzustellen und andere anwendungsspezifische Einstellungen zu konfigurieren.

Weitere Informationen und Beispielskripts für die einzelnen Schritte in diesem Thema finden Sie unterAutomatisieren Sie die Anwendungsbereitstellung.

Verwalten von Anwendungen in der AWS Panorama Panorama-Konsole

Verwenden Sie die AWS Panorama Panorama-Konsole, um bereitgestellte Anwendungen zu verwalten.

Abschnitte

Eine Anwendung aktualisieren oder kopieren

Um eine Anwendung zu aktualisieren, verwenden Sie dieErsetzen-Option. Wenn Sie eine Anwendung ersetzen, können Sie deren Code oder Modelle aktualisieren.

So aktualisieren Sie eine Anwendung

1. Öffnen der AWS Panorama Panorama-KonsoleBereitgestellte Anwendungsseite.
2. Wählen Sie eine Anwendung aus.
3. Wählen Sie Replace (Ersetzen) aus.
4. Folgen Sie den Anweisungen, um eine neue Version oder Anwendung zu erstellen.

Es gibt auch eineKlonOption, die sich ähnlich verhält wieErsetzen, entfernt aber nicht die alte Version der Anwendung. Sie können diese Option verwenden, um Änderungen an einer Anwendung zu testen, ohne die ausgeführte Version anzuhalten, oder um eine Version, die Sie bereits gelöscht haben, erneut bereitzustellen.

Versionen und Anwendungen löschen

Um nicht verwendete Anwendungsversionen zu bereinigen, löschen Sie sie aus Ihren Appliances.

So löschen Sie eine Anwendung

1. Öffnen der AWS Panorama Panorama-KonsoleBereitgestellte Anwendungsseite.
2. Wählen Sie eine Anwendung aus.
3. WählenLöschen vom Gerät.

Verpackungskonfiguration

Wenn Sie den Befehl AWS Panorama Application CLI-Befehl verwenden`panorama-cli package-application`lädt die CLI die Assets Ihrer Anwendung auf Amazon S3 hoch und registriert sie bei AWS Panorama. Zu den Assets gehören Binärdateien (Container-Images und Modelle) und Deskriptordateien, die die AWS Panorama Appliance während der Bereitstellung herunterlädt. Um die Assets eines Pakets zu registrieren, stellen Sie eine separate Paketkonfigurationsdatei zur Verfügung, die das Paket, seine Assets und seine Schnittstelle definiert.

Das folgende Beispiel zeigt eine Paketkonfiguration für einen Codeknoten mit einer Eingabe und einer Ausgabe. Der Videoeingang bietet Zugriff auf Bilddaten aus einem Kamerastream. Der Ausgabeknoten sendet verarbeitete Bilder an ein Display.

Example Pakete/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"
                    }
                ]
            }
        ]
    }
}
```

Die`assets`gibt die Namen der Artefakte an, die die AWS Panorama Application CLI auf Amazon S3 hochgeladen hat. Wenn Sie eine Beispielanwendung oder eine Anwendung von einem anderen Benutzer importieren, kann dieser Abschnitt leer sein oder sich auf Assets beziehen, die sich nicht in Ihrem Konto befinden. Wenn du rennst`panorama-cli package-application`füllt die AWS Panorama Application CLI diesen Abschnitt mit den richtigen Werten.

Das Manifest der AWS-Panorama-Anwendung

Wenn Sie eine Anwendung bereitstellen, geben Sie eine Konfigurationsdatei an, die als Anwendungsmanifest bezeichnet wird. Diese Datei definiert die Anwendung als Diagramm mit Knoten und Kanten. Das Anwendungsmanifest ist Teil des Anwendungsquellcodes und wird im`graphs`-Verzeichnis.

Example diagramm/aws-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"
            }
        ]
    }
}
```

Knoten sind durch Kanten verbunden, die Zuordnungen zwischen den Ein- und Ausgängen von Knoten angeben. Die Ausgabe eines Knotens stellt eine Verbindung mit der Eingabe eines anderen her und bildet einen Graphen.

JSON-Schema

Das Format von Anwendungsmanifest- und Override-Dokumenten ist in einem JSON-Schema definiert. Sie können das JSON-Schema verwenden, um Ihre Konfigurationsdokumente vor der Bereitstellung zu überprüfen. Das JSON-Schema ist in diesem Handbuch verfügbarGitHub-Repository.

- JSON-Schema–aws-panorama-developer-Guide/Ressourcen

Anwendungs-Knoten

Knoten sind Modelle, Code, Kamerastreams, Ausgabe und Parameter. Ein Knoten hat eine Schnittstelle, die seine Ein- und Ausgänge definiert. Die Schnittstelle kann in einem Paket in Ihrem Konto, einem von AWS Panorama bereitgestellten Paket oder einem integrierten Typ definiert werden.

Im folgenden Beispiel`code_node`und`model_node`beziehen Sie sich auf den Beispielcode und die Modellpakete, die in der Beispielanwendung enthalten sind.`camera_node`verwendet ein von AWS Panorama bereitgestelltes Paket, um einen Platzhalter für einen Kamera-Stream zu erstellen, den Sie während der Bereitstellung angeben.

Example graph.json — Knoten

```
        "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

Kanten ordnen die Ausgabe von einem Knoten der Eingabe eines anderen zu. Im folgenden Beispiel ordnet der erste Edge die Ausgabe eines Kamera-Stream-Knotens der Eingabe eines Anwendungscode-Knotens zu. Die Namen`video_in`und`video_out`sind in den Schnittstellen der Knotenpakete definiert.

Example graph.json — Kanten

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

In Ihrem Anwendungscode können Sie den`inputs`und`outputs`-Attribute, um Bilder aus dem Eingabestream zu erhalten und Bilder an den Ausgabestream zu senden.

Example application.py — Videoeingang und -ausgabe

```
    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)
```

Abstract Nodes

In einem Anwendungsmanifest bezieht sich ein abstrakter Knoten auf ein von AWS Panorama definiertes Paket, das Sie als Platzhalter in Ihrem Anwendungsmanifest verwenden können. AWS Panorama bietet zwei Arten von abstraktem Knoten.

- Kamera-Stream— Wählen Sie den Kamera-Stream aus, den die Anwendung während der Bereitstellung verwendet.

  Package name–`panorama::abstract_rtsp_media_source`

  Schnittstellenname–`rtsp_v1_interface`

- HDMI-Ausgang— Zeigt an, dass die Anwendung Video ausgibt.

  Package name–`panorama::hdmi_data_sink`

  Schnittstellenname–`hdmi0`

Das folgende Beispiel zeigt einen grundlegenden Satz von Paketen, Knoten und Kanten für eine Anwendung, die Kamerastreams verarbeitet und Video an ein Display ausgibt. Der Kameraknoten, der die Schnittstelle von der`abstract_rtsp_media_source`Paket in AWS Panorama, kann mehrere Kamerastreams als Eingabe akzeptieren. Der Ausgabeknoten, der referenziert`hdmi_data_sink`, gewährt Anwendungscode Zugriff auf einen Videopuffer, der vom HDMI-Anschluss der Appliance ausgegeben wird.

Example graph.json — Abstrakte Knoten

```
{
    "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"
            }
        ]
    }
}
```

Anwendungsparameter

Parameter sind Knoten, die einen Basistyp haben und während der Bereitstellung überschrieben werden können. Ein Parameter kann einen Standardwert und einenDekorateur, das den Benutzer der Anwendung anweist, wie er sie konfiguriert.

Parametertypen

- `string`— Eine Zeichenfolge. Zum Beispiel `DEBUG`.
- `int32`— Eine ganze Zahl. Beispiel, `20`
- `float32`— Eine Gleitkommazahl. Beispiel, `47.5`
- `boolean`–`true`oder`false`eaus.

Das folgende Beispiel zeigt zwei Parameter, eine Zeichenfolge und eine Zahl, die als Eingaben an einen Codeknoten gesendet werden.

Example graph.json — Parameter

```
       "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"
            }
            ...
        ]
    }
```

Sie können Parameter direkt im Anwendungsmanifest ändern oder zur Bereitstellungszeit neue Werte mit Überschreibungen angeben. Weitere Informationen finden Sie unter Bereitstellungszeitkonfiguration mit Overrides.

Bereitstellungszeitkonfiguration mit Overrides

Sie konfigurieren Parameter und abstrakte Knoten während der Bereitstellung. Wenn Sie die AWS Panorama Panorama-Konsole zum Bereitstellen verwenden, können Sie für jeden Parameter einen Wert angeben und einen Kamerastream als Eingabe auswählen. Wenn Sie die AWS Panorama API zum Bereitstellen von Anwendungen verwenden, geben Sie diese Einstellungen mit einem Overrides-Dokument an.

Ein überschreibendes Dokument ähnelt in der Struktur einem Anwendungsmanifest. Für Parameter mit Basistypen definieren Sie einen Knoten. Für Kamerastreams definieren Sie einen Knoten und ein Paket, das einem registrierten Kamera-Stream zugeordnet ist. Dann definieren Sie eine Überschreibung für jeden Knoten, der den Knoten aus dem Anwendungsmanifest angibt, den er ersetzt.

Example überschreibt. 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"
    }
}
```

Im vorhergehenden Beispiel definiert das Dokument Überschreibungen für einen Zeichenfolgenparameter und einen abstrakten Kameraknoten. Die`node0verrides`teilt AWS Panorama mit, welche Knoten in diesem Dokument welche im Anwendungsmanifest überschreiben.

ErstellungAWS PanoramaAnwendungen

Anwendungen werden auf demAWS PanoramaAppliance zur Durchführung von Bildverarbeitungsaufgaben in Videostreams. Sie können Computer Vision-Anwendungen erstellen, indem Sie Python-Code- und Machine-Learning-Modelle kombinieren und sie in derAWS PanoramaAppliance über das Internet. Anwendungen können Videos an eine Anzeige senden oder das AWS-SDK verwenden, um Ergebnisse an AWS-Services zu senden.

EINModellanalysiert Bilder, um Personen, Fahrzeuge und andere Objekte zu erkennen. Basierend auf Bildern, die es während des Trainings gesehen hat, sagt Ihnen das Modell, was es für etwas hält und wie sicher es ist, etwas zu erraten. Sie können Modelle mit Ihren eigenen Bilddaten trainieren oder mit einer Probe beginnen.

Die Anwendung istCodeverarbeitet Standbilder aus einem Kamerastream, sendet sie an ein Modell und verarbeitet das Ergebnis. Ein Modell erkennt möglicherweise mehrere Objekte und gibt deren Form und Position zurück. Der Code kann diese Informationen verwenden, um dem Video Text oder Grafiken hinzuzufügen oder um Ergebnisse an eineAWSService für die Lagerung oder Weiterverarbeitung.

Um Bilder aus einem Stream zu erhalten, mit einem Modell zu interagieren und Videos auszugeben, verwendet der AnwendungscodedieAWS PanoramaAnwendungs-SDKaus. Das Anwendungs-SDK ist eine Python-Bibliothek, die Modelle unterstützt, die mit PyTorch, Apache MXNet und TensorFlowaus.

Themen

Computer Vision-Modelle

Ein Computer Vision Model ist ein Softwareprogramm, das darauf trainiert ist, Objekte in Bildern zu erkennen. Ein Modell lernt, eine Reihe von Objekten zu erkennen, indem es zunächst Bilder dieser Objekte durch Training analysiert. Ein Computer Vision-Modell verwendet ein Bild als Eingabe und gibt Informationen über die Objekte aus, die es erkennt, z. B. den Objekttyp und seine Position. AWS Panorama unterstützt Computer Vision-ModellePyTorch, die mit Apache MXNet und TensorFlow erstellt wurden.

> **Note**
>
> Eine Liste der vorgefertigten Modelle, die mit AWS Panorama getestet wurden, finden Sie unter Modellkompatibilität.

Abschnitte

Modelle im Code verwenden

Ein Modell gibt ein oder mehrere Ergebnisse zurück, die Wahrscheinlichkeiten für erkannte Klassen, Ortsinformationen und andere Daten beinhalten können. Das folgende Beispiel zeigt, wie Inferenz auf ein Bild aus einem Videostream ausgeführt und die Ausgabe des Modells an eine Verarbeitungsfunktion gesendet wird.

Example application.py — Inferenz

```
    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)
```

Das folgende Beispiel zeigt eine Funktion, die Ergebnisse aus dem grundlegenden Klassifikationsmodell verarbeitet. Das Stichprobenmodell gibt eine Reihe von Wahrscheinlichkeiten zurück. Dies ist der erste und einzige Wert im Ergebnisarray.

Example application.py — Ergebnisse werden verarbeitet

```
    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)
```

Der Anwendungscode findet die Werte mit den höchsten Wahrscheinlichkeiten und ordnet sie Labels in einer Ressourcendatei zu, die während der Initialisierung geladen wird.

Ein benutzerdefiniertes Modell erstellen

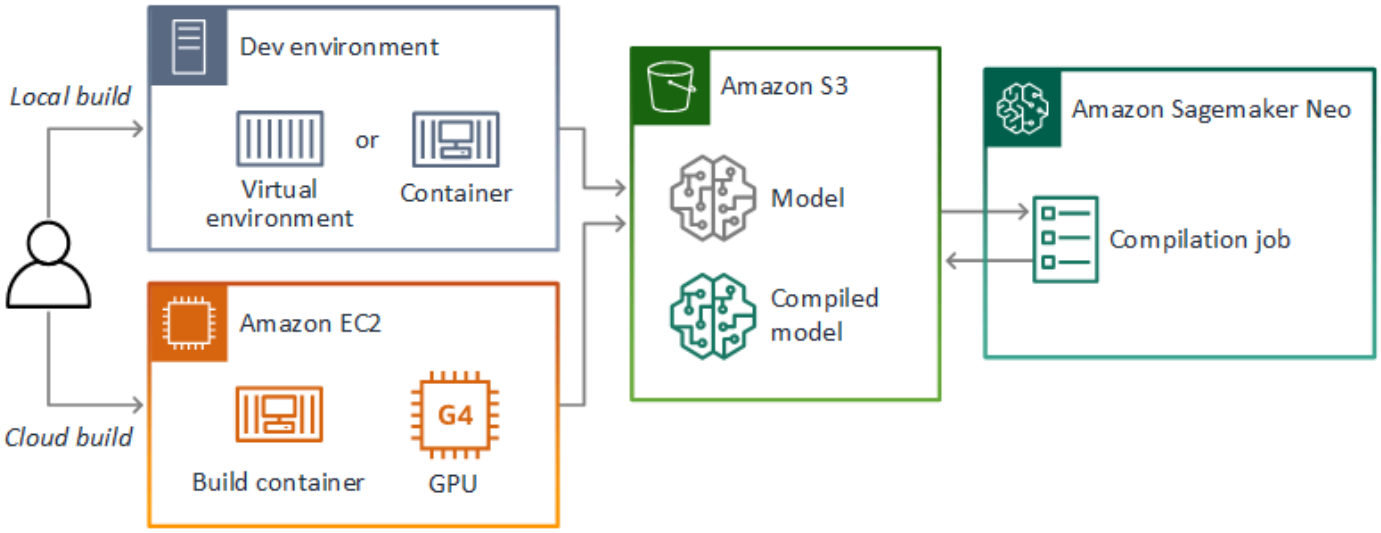
Sie können Modelle verwenden, die Sie integriert habenPyTorch, Apache MXNet und TensorFlow in AWS Panorama-Anwendungen. Als Alternative zum Erstellen und Trainieren von Modellen in SageMaker können Sie ein trainiertes Modell verwenden oder Ihr eigenes Modell mit einem

unterstützten Framework erstellen und trainieren und es in eine lokale Umgebung oder in Amazon EC2 exportieren.

> **Note**
>
> Einzelheiten zu den von SageMaker Neo unterstützten Framework-Versionen und Dateiformaten finden Sie unter Unterstützte Frameworks im Amazon SageMaker Developer Guide.

Das Repository für dieses Handbuch enthält eine Beispielanwendung, die diesen Arbeitsablauf für ein Keras-Modell im TensorFlow `SavedModel` Format demonstriert. Es verwendet TensorFlow 2 und kann lokal in einer virtuellen Umgebung oder in einem Docker-Container ausgeführt werden. Die Beispiel-App enthält auch Vorlagen und Skripts für die Erstellung des Modells auf einer Amazon EC2-Instance.

- Beispielanwendung für ein benutzerdefiniertes Modell



AWS Panorama verwendet SageMaker Neo, um Modelle für die Verwendung auf der AWS Panorama Appliance zu kompilieren. Verwenden Sie für jedes Framework das von SageMaker Neo unterstützte Format und packen Sie das Modell in ein `.tar.gz` Archiv.

Weitere Informationen finden Sie unter Kompilieren und Bereitstellen von Modellen mit Neo im Amazon SageMaker Developer Guide.

Ein Modell verpacken

Ein Modellpaket umfasst einen Deskriptor, eine Paketkonfiguration und ein Modellarchiv. Wie in einem Anwendungs-Image-Paket teilt die Paketkonfiguration dem AWS Panorama-Service mit, wo das Modell und der Deskriptor in Amazon S3 gespeichert sind.

Example /Pakete/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**
>
> Geben Sie nur die Haupt- und Nebenversion der Framework-Version an. Eine Liste der unterstützten PyTorch Versionen von Apache MXNet und TensorFlow Versionen finden Sie unter Unterstützte Frameworks.

Verwenden Sie den `import-raw-model` CLI-Befehl AWS Panorama Application, um ein Modell zu importieren. Wenn Sie Änderungen am Modell oder seinem Deskriptor vornehmen, müssen Sie diesen Befehl erneut ausführen, um die Assets der Anwendung zu aktualisieren. Weitere Informationen finden Sie unter Änderung des Computer Vision-Modells.

Das JSON-Schema der Deskriptordatei finden Sie unter assetDescriptor.schema.json.

Trainingsmodelle

Wenn Sie ein Modell trainieren, verwenden Sie Bilder aus der Zielumgebung oder aus einer Testumgebung, die der Zielumgebung sehr ähnlich ist. Berücksichtigen Sie die folgenden Faktoren, die die Modellleistung beeinflussen können:

- Beleuchtung — Die Lichtmenge, die von einem Objekt reflektiert wird, bestimmt, wie viele Details das Modell analysieren muss. Ein Modell, das mit Bildern von gut beleuchteten Motiven trainiert wurde, funktioniert in Umgebungen mit wenig Licht oder Gegenlicht möglicherweise nicht gut.
- Auflösung — Die Eingabegröße eines Modells ist in der Regel auf eine Auflösung zwischen 224 und 512 Pixeln in einem quadratischen Seitenverhältnis festgelegt. Bevor Sie ein Videobild an das Modell weitergeben, können Sie es verkleinern oder auf die erforderliche Größe zuschneiden.
- Bildverzerrung — Die Brennweite und die Linsenform einer Kamera können dazu führen, dass Bilder von der Bildmitte weg verzerrt sind. Die Position einer Kamera bestimmt auch, welche Merkmale eines Motivs sichtbar sind. Eine Overhead-Kamera mit einem Weitwinkelobjektiv zeigt beispielsweise die Oberseite eines Motivs, wenn es sich in der Bildmitte befindet, und eine schiefe Ansicht der Seite des Motivs, wenn es sich weiter von der Bildmitte entfernt.

Um diese Probleme zu lösen, können Sie Bilder vorverarbeiten, bevor Sie sie an das Modell senden, und das Modell anhand einer größeren Anzahl von Bildern trainieren, die Abweichungen in realen Umgebungen widerspiegeln. Wenn ein Modell in Lichtsituationen und mit einer Vielzahl von Kameras arbeiten muss, benötigen Sie mehr Daten für das Training. Sie können nicht nur mehr Bilder sammeln, sondern auch mehr Trainingsdaten erhalten, indem Sie Variationen Ihrer vorhandenen Bilder erstellen, die schief oder unterschiedlich beleuchtet sind.

Ein Anwendungs-Image erstellen

Die AWS Panorama Appliance führt Anwendungen als Container-Dateisysteme aus, die aus einem von Ihnen erstellten Image exportiert wurden. Sie geben die Abhängigkeiten und Ressourcen Ihrer Anwendung in einem Dockerfile an, das das Basisimage der AWS Panorama-Anwendung als Ausgangspunkt verwendet.

Um ein Anwendungs-Image zu erstellen, verwenden Sie Docker und die AWS Panorama Application CLI. Das folgende Beispiel aus der Beispielanwendung dieses Handbuchs veranschaulicht diese Anwendungsfälle.

Example /Pakete/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
```

Die folgenden Dockerfile-Anweisungen werden verwendet.

- `FROM`— Lädt das Basis-Image der Anwendung (`public.ecr.aws/panorama/panorama-application`).
- `WORKDIR`— Legt das Arbeitsverzeichnis für das Bild fest. `/panorama`wird für Anwendungscode und zugehörige Dateien verwendet. Diese Einstellung bleibt nur während des Builds bestehen und wirkt sich nicht auf das Arbeitsverzeichnis Ihrer Anwendung zur Laufzeit aus (`/`).
- `COPY`— Kopiert Dateien von einem lokalen Pfad in einen Pfad auf dem Bild. `COPY . .`kopiert die Dateien im aktuellen Verzeichnis (das Paketverzeichnis) in das Arbeitsverzeichnis des Images. Beispielsweise wird der Anwendungscode von `packages/123456789012-SAMPLE_CODE-1.0/application.py` nach kopiert`/panorama/application.py`.
- `RUN`— Führt während des Builds Shell-Befehle auf dem Image aus. Ein einziger `RUN` Vorgang kann mehrere Befehle nacheinander ausführen, indem er && zwischen Befehlen verwendet wird. In diesem Beispiel wird der `pip` Paketmanager aktualisiert und anschließend die in `requirements.txt` aufgelisteten Bibliotheken installiert.

Sie können andere Anweisungen wie `ADD` und verwenden`ARG`, die bei der Erstellung nützlich sind. Anweisungen, die dem Container Laufzeitinformationen hinzufügen, wie z. B.`ENV`, funktionieren nicht

mit AWS Panorama. AWS Panorama führt keinen Container aus dem Image aus. Es verwendet das Image nur, um ein Dateisystem zu exportieren, das auf die Appliance übertragen wird.

Angeben von Abhängigkeiten

`requirements.txt`ist eine Python-Anforderungsdatei, die die von der Anwendung verwendeten Bibliotheken angibt. Die Beispielanwendung verwendet Open CV und dieAWS SDK for Python (Boto3).

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

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

Der `pip install` Befehl im Dockerfile installiert diese Bibliotheken in das `dist-packages` Python-Verzeichnis unter`/usr/local/lib`, sodass sie von Ihrem Anwendungscode importiert werden können.

Lokaler Speicher

AWS Panorama reserviert das `/opt/aws/panorama/storage` Verzeichnis für die Speicherung von Anwendungen. Ihre Anwendung kann in diesem Pfad Dateien erstellen und ändern. Dateien, die im Speicherverzeichnis erstellt wurden, bleiben auch bei Neustarts erhalten. Andere temporäre Dateispeicherorte werden beim Booten gelöscht.

Bild-Assets erstellen

Wenn Sie mit der AWS Panorama Application CLI ein Image für Ihr Anwendungspaket erstellen, wird die CLI `docker build` im Paketverzeichnis ausgeführt. Dadurch wird ein Anwendungs-Image erstellt, das Ihren Anwendungscode enthält. Die CLI erstellt dann einen Container, exportiert sein Dateisystem, komprimiert es und speichert es in dem Ordner. `assets`

```
$ 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
```

Der JSON-Block in der Ausgabe ist eine Asset-Definition, die die CLI zur Paketkonfiguration (`package.json`) hinzufügt und beim AWS Panorama-Service registriert. Die CLI kopiert auch die Deskriptordatei, die den Pfad zum Anwendungsskript (dem Einstiegspunkt der Anwendung) angibt.

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

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

Im Assets-Ordner sind der Deskriptor und das Anwendungs-Image nach ihrer SHA-256-Prüfsumme benannt. Dieser Name wird als eindeutige Kennung für das Asset verwendet, wenn es in Amazon S3 gespeichert wird.

AWS-Services von Ihrem Anwendungscode aus aufrufen

Sie können den verwendenAWS SDK for Python (Boto), um AWS-Services von Ihrem Anwendungscode aus aufzurufen. Wenn Ihr Modell beispielsweise etwas Ungewöhnliches entdeckt, können Sie Metriken an Amazon posten CloudWatch, eine Benachrichtigung mit Amazon SNS senden, ein Bild in Amazon S3 speichern oder eine Lambda-Funktion zur weiteren Verarbeitung aufrufen. Die meisten AWS-Services verfügen über eine öffentliche API, die Sie mit dem AWS SDK verwenden können.

Die Appliance hat standardmäßig keine Berechtigung, auf AWS-Services zuzugreifen. Um ihr die Berechtigung zu erteilen, erstellen Sie eine Rolle für die Anwendung und weisen Sie sie während der Bereitstellung der Anwendungsinstanz zu.

Abschnitte

- Verwenden von Amazon S3
- DasAWS IoT MQTT-Thema verwenden

Verwenden von Amazon S3

Sie können mit Amazon S3 Verarbeitungsergebnisse und andere Anwendungsdaten speichern.

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

DasAWS IoT MQTT-Thema verwenden

Sie können mit Amazon SDK for Python (Boto3) Nachrichten an ein MQTT-Thema inAWS IoT. Im folgenden Beispiel veröffentlicht die Anwendung Beiträge zu einem Thema, das nach dem Dingnamen der Appliance benannt ist, den Sie in der AWS IoTKonsole finden.

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

Wählen Sie einen Namen, der die Geräte-ID oder eine andere Kennung Ihrer Wahl angibt. Um Nachrichten zu veröffentlichen, benötigt die Anwendung die Berechtigung zum Aufrufen`iot:Publish`.

Um eine MQTT-Warteschlange zu überwachen

1. Öffnen Sie die Testseite derAWS IoT Konsole.
2. Geben Sie für Abonnementthema den Namen des Themas ein. Zum Beispiel `panorama/panorama_my-appliance_Thing_a01e373b`.
3. Wählen Sie Thema abonnieren aus.

Das AWS Panorama Application SDK

Das AWS Panorama Application SDK ist eine Python-Bibliothek zur Entwicklung von AWS Panorama Panorama-Anwendungen. In deinerAnwendungscodeverwenden, verwenden Sie das AWS Panorama Application SDK, um ein Computer-Vision-Modell zu laden, Inferenz auszuführen und Video an einen Monitor auszugeben.

> **Note**
>
> Um sicherzustellen, dass Sie Zugriff auf die neuesten Funktionen des AWS Panorama Application SDK-Daten haben,Aktualisieren Sie die Appliance-Softwareaus.

Weitere Informationen zu den Klassen, die das Anwendungs-SDK definiert, und zu ihren Methoden finden Sie unterReferenz des Anwendungs-SDK-Objektsaus.

Abschnitte

- Hinzufügen von Text und Feldern zur Ausgabe von Video

Hinzufügen von Text und Feldern zur Ausgabe von Video

Mit dem AWS Panorama SDK können Sie einen Videostream auf ein Display ausgeben. Das Video kann Text und Felder enthalten, die die Ausgabe des Modells, den aktuellen Status der Anwendung oder andere Daten anzeigen.

Jedes Objekt im`video_in`array ist ein Bild aus einem Kamera-Stream, der mit der Appliance verbunden ist. Der Typ dieses Objekts ist`panoramasdk.media`aus. Es verfügt über Methoden zum Hinzufügen von Text- und rechteckigen Feldern zum Bild, die Sie dann dem`video_out`Array.

Im folgenden Beispiel fügt die Beispielanwendung für jedes Ergebnis eine Beschriftung hinzu. Jedes Ergebnis wird an derselben linken Position, jedoch in verschiedenen Höhen positioniert.

```
        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)
```

Um dem Ausgabebild ein Feld hinzuzufügen, verwenden Sie`add_rect`aus. Diese Methode benötigt 4 Werte zwischen 0 und 1 und gibt die Position der oberen linken und unteren rechten Ecke des Feldes an.

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

Ausführen mehrerer Threads

Sie können Ihre Anwendungslogik in einem Verarbeitungsthread ausführen und andere Threads für andere Hintergrundprozesse verwenden. Sie können beispielsweise einen Thread-Prinzip erstellen, derbedient HTTP-Datenverkehrzum Debuggen, oder ein Thread, der Inferenzergebnisse überwacht und Daten anAWSaus.

Um mehrere Threads auszuführen, verwenden Sie denThreading-Modulaus der Python-Standardbibliothek, um für jeden Prozess einen Thread zu erstellen. Das folgende Beispiel zeigt die Hauptschleife der Debugserver-Beispielanwendung, die ein Anwendungsobjekt erstellt und zum Ausführen verwendetdreiThreads.

Example Pakete/123456789012-debug_server-1.0/application.py— Hauptschleife

```
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.')
```

Wenn alle Threads beendet werden, startet die Anwendung von selbst neu. Die`run_cv`loop verarbeitet Bilder aus Kamerastreams. Wenn es ein Signal zum Stoppen erhält, fährt es den

Debugger-Prozess herunter, der einen HTTP-Server ausführt und sich nicht selbst herunterfahren kann. Jeder Thread muss seine eigenen Fehler behandeln. Wenn ein Fehler nicht abgefangen und protokolliert wird, wird der Thread im Hintergrund beendet.

Example Pakete/123456789012-debug_server-1.0/application.py— Verarbeitungsschleife

```
    # 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")
```

Threads kommunizieren über die`self`-Objekt. Um die Anwendungsverarbeitungsschleife neu zu starten, ruft der Debugger-Threads`stop`-Methode. Diese Methode setzt eine`terminate`-Attribut, das den anderen Threads signalisiert, herunterzufahren.

Example Pakete/123456789012-debug_server-1.0/application.py— Stoppen -Methode

```
    # 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)
```

Serving von Datenverkehr

Sie können Anwendungen lokal überwachen oder debuggen, indem Sie neben Ihrem Anwendungscode einen HTTP-Server ausführen. Um externen Datenverkehr zu bedienen, ordnen Sie Ports auf der AWS Panorama Appliance Ports in Ihrem Anwendungscontainer zu.

> **Important**
>
> Standardmäßig akzeptiert die AWS Panorama Panorama-Appliance keinen eingehenden Datenverkehr an irgendwelchen Ports. Das Öffnen von Ports auf der Appliance birgt ein implizites Sicherheitsrisiko. Wenn Sie diese Funktion verwenden, müssen Sie zusätzliche Schritte ausführen, umschützen Sie Ihre Appliance vor externem Datenverkehrund sichere Kommunikation zwischen autorisierten Clients und der Appliance.
> Der in diesem Handbuch enthaltene Beispielcode dient zu Demonstrationszwecken und implementiert keine Authentifizierung, Autorisierung oder Verschlüsselung.

Sie können Ports im Bereich von 8000 bis 9000 auf der Appliance öffnen. Wenn diese Ports geöffnet sind, können sie Datenverkehr von jedem routingfähigen Client empfangen. Wenn Sie Ihre Anwendung bereitstellen, geben Sie an, welche Ports geöffnet werden sollen, und ordnen Ports auf der Appliance Ports in Ihrem Anwendungscontainer zu. Die Appliance-Software leitet den Datenverkehr an den Container weiter und sendet Antworten an den Anforderer zurück. Anfragen werden an dem von Ihnen angegebenen Appliance-Port empfangen, und die Antworten werden an einem zufälligen kurzlebigen Port ausgegeben.

Konfigurieren von Ports eingehender Abfragen

Sie geben Portzuordnungen an drei Stellen in Ihrer Anwendungskonfiguration an. Das Codepaket ist`package.json`verwenden, geben Sie den Port an, den der Codeknoten in einem`network`Block. Das folgende Beispiel erklärt, dass der Knoten auf Port 80 lauscht.

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

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

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

Im Anwendungsmanifest deklarieren Sie eine Routing-Regel, die einen Port auf der Appliance einem Port im Codecontainer der Anwendung zuordnet. Im folgenden Beispiel wird eine Regel hinzugefügt, die Port 8080 auf dem Gerät Port 80 auf der`code_node`Container.

Example graphs/my-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."
            }
        }
    ]
```

Wenn Sie die Anwendung bereitstellen, geben Sie dieselben Regeln in der AWS Panorama Panorama-Konsole oder mit einem Override-Dokument an, das an dieCreateApplicationInstanceAPI. Sie müssen diese Konfiguration bei der Bereitstellung angeben, um zu bestätigen, dass Sie Ports auf der Appliance öffnen möchten.

Example graphs/meine-app/override.json

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

Wenn der im Anwendungsmanifest angegebene Geräte-Port von einer anderen Anwendung verwendet wird, können Sie das Override-Dokument verwenden, um einen anderen Port auszuwählen.

Serving Datenverkehr

Wenn Ports auf dem Container geöffnet sind, können Sie einen Socket öffnen oder einen Server ausführen, um eingehende Anfragen zu bearbeiten. Die`debug-server`Beispiel zeigt eine grundlegende Implementierung eines HTTP-Servers, der zusammen mit Computer Vision-Anwendungscode ausgeführt wird.

> **Important**
>
> Die Beispielimplementierung ist für die Produktion nicht sicher. Um zu vermeiden, dass Ihre Appliance anfällig für Angriffe wird, müssen Sie geeignete Sicherheitskontrollen in Ihrem Code und Ihrer Netzwerkkonfiguration implementieren.

Example Pakete/123456789012-debug_server-1.0/application.py— HTTP-Server

```
    # 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.')
```

Der Server akzeptiert GET-Anfragen auf der`/status`-Pfad, um einige Informationen über die Anwendung abzurufen. Es akzeptiert auch eine POST-Anfrage an`/restart`um die Anwendung neu zu starten.

Um diese Funktionalität zu demonstrieren, führt die Beispielanwendung einen HTTP-Client in einem separaten Thread aus. Der Kunde ruft den`/status`-Pfad kurz nach dem Start über das lokale Netzwerk und startet die Anwendung einige Minuten später neu.

Example Pakete/123456789012-debug_server-1.0/application.py— HTTP-Clients

```
    # 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")
```

Die Hauptschleife verwaltet die Threads und startet die Anwendung neu, wenn sie beendet werden.

Example Pakete/123456789012-debug_server-1.0/application.py— Hauptschleife

```
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.')
```

Informationen zur Bereitstellung der Beispielanwendung finden Sie in derAnweisungen in diesem Handbuch GitHub Repository.

Verwendung der GPU

Sie können auf den Grafikprozessor (GPU) der AWS Panorama Appliance zugreifen, um GPU-beschleunigte Bibliotheken zu verwenden oder Modelle für maschinelles Lernen in Ihrem Anwendungscode auszuführen. Um den GPU-Zugriff zu aktivieren, fügen Sie der Paketkonfiguration GPU-Zugriff als Anforderung hinzu, nachdem Sie Ihren Anwendungscode-Container erstellt haben.

> **Important**
>
> Wenn Sie den GPU-Zugriff aktivieren, können Sie Modellknoten in keiner Anwendung auf der Appliance ausführen. Aus Sicherheitsgründen ist der GPU-Zugriff eingeschränkt, wenn die Appliance ein mit SageMaker Neo kompiliertes Modell ausführt. Mit GPU-Zugriff müssen Sie Ihre Modelle in Anwendungscodeknoten ausführen, und alle Anwendungen auf dem Gerät teilen sich den Zugriff auf die GPU.

Um den GPU-Zugriff für Ihre Anwendung zu aktivieren, aktualisieren Sie die Paketkonfiguration, nachdem Sie das Paket mit der AWS Panorama-Anwendungs-CLI erstellt haben. Das folgende Beispiel zeigt den `requirements` Block, der dem Anwendungscodeknoten GPU-Zugriff hinzufügt.

Example package.json mit Anforderungsblock

```
{
    "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": [
    ...
```

Aktualisieren Sie die Paketkonfiguration zwischen den Build- und Paketierungsschritten in Ihrem Entwicklungsworkflow.

So stellen Sie eine Anwendung mit GPU-Zugriff bereit

1. Verwenden Sie den `build-container` Befehl, um den Anwendungscontainer zu erstellen.

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

2. Fügen Sie den `requirements` Block zur Paketkonfiguration hinzu.
3. Verwenden Sie den `package-application` Befehl, um das Container-Asset und die Paketkonfiguration hochzuladen.

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

4. Stellen Sie die Anwendung bereit.

Beispielanwendungen, die GPU-Zugriff verwenden, finden Sie im aws-panorama-samplesGitHubRepository.

Einrichten einer Entwicklungsumgebung in Windows

Um eine AWS Panorama Panorama-Anwendung zu erstellen, verwenden Sie Docker, Befehlszeilentools und Python. In Windows können Sie eine Entwicklungsumgebung einrichten, indem Sie Docker Desktop mit Windows Subsystem für Linux und Ubuntu verwenden. Dieses Tutorial führt Sie durch den Einrichtungsprozess für eine Entwicklungsumgebung, die mit AWS Panorama Panorama-Tools und Beispielanwendungen getestet wurde.

Abschnitte

Voraussetzungen

Um diesem Tutorial zu folgen, benötigen Sie eine Version von Windows, die das Windows-Subsystem für Linux 2 (WSL 2) unterstützt.

- Windows 10 Version 1903 und höher (Build 18362 und höher) oder Windows 11
- Windows-Funktionen
  - Windows-Subsystem für Linux
  - Hyper-V
  - Plattform für virtuelle Maschinen

Dieses Tutorial wurde mit den folgenden Softwareversionen entwickelt.

- Ubuntu 20.04
- Python 3.8.5
- Docker 20.10.8

Installieren Sie WSL 2 und Ubuntu

Wenn Sie Windows 10 Version 2004 und höher (Build 19041 und höher) haben, können Sie WSL 2 und Ubuntu 20.04 mit dem folgenden PowerShell-Befehl installieren.

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

Befolgen Sie für ältere Windows-Versionen die Anweisungen in der WSL 2-Dokumentation: Manuelle Installationsschritte für ältere Versionen

Docker-Installation

Um Docker Desktop zu installieren, laden Sie das Installerpaket herunter und führen Sie es aushub.docker.comaus. Wenn Probleme auftreten, folgen Sie den Anweisungen auf der Docker-Website: Docker Desktop WSL 2 Backendaus.

Führen Sie Docker Desktop aus und folgen Sie dem Tutorial zum ersten Ausführen, um einen Beispielcontainer zu erstellen.

> **Note**
>
> Docker Desktop aktiviert Docker nur in der Standardverteilung. Wenn Sie vor dem Ausführen dieses Tutorials andere Linux-Distributionen installiert haben, aktivieren Sie Docker in der neu installierten Ubuntu-Distribution im Docker Desktop-Einstellungsmenü unterRessourcen,WSL-Integrationaus.

Konfigurieren von Ubuntu

Sie können jetzt Docker-Befehle in Ihrer virtuellen Maschine von Ubuntu ausführen. Um ein Befehlszeilenterminal zu öffnen, führen Sie die Verteilung über das Startmenü aus. Wenn Sie es zum ersten Mal ausführen, konfigurieren Sie einen Benutzernamen und ein Kennwort, mit denen Sie Administratorbefehle ausführen können.

Aktualisieren Sie die Software und die Installationstools der virtuellen Maschine, um die Konfiguration Ihrer Entwicklungsumgebung abzuschließen.

So konfigurieren Sie die virtuelle Maschine

1. Aktualisieren Sie die Software, die mit Ubuntu geliefert wird.

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

2. Installieren Sie Entwicklungstools mit apt.

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

3. Installieren Sie Python-Bibliotheken mit pip.

```
$ pip3 install awscli panoramacli
```

4. Öffnen Sie ein neues Terminal und führen Sie dann`aws configure`um denAWS CLIaus.

```
$ aws configure
```

   Wenn Sie noch keine Zugriffsschlüssel besitzen, können Sie diese imIAM-Konsoleaus.

Laden Sie abschließend die Beispielanwendung herunter und importieren Sie sie.

So erhalten Sie die Beispielanwendung

1. Downloaden und extrahieren Sie die Beispielanwendung.

```
$ 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. Führen Sie die enthaltenen Skripte aus, um die Kompilierung zu testen, den Anwendungscontainer zu erstellen und Pakete in AWS Panorama hochzuladen.

```
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
```

Die AWS Panorama Application CLI lädt Pakete hoch und registriert sie beim AWS Panorama Panorama-Service. Sie können jetztBereitstellen der Beispielanwendungmit der AWS Panorama Panorama-Konsole.

Nächste Schritte

Um die Projektdateien zu untersuchen und zu bearbeiten, können Sie den Datei-Explorer oder eine integrierte Entwicklungsumgebung (IDE) verwenden, die WSL unterstützt.

Um auf das Dateisystem der virtuellen Maschine zuzugreifen, öffnen Sie den Datei-Explorer und geben Sie`\\wsl$`In der Navigationsleiste. Dieses Verzeichnis enthält einen Link zum Dateisystem der virtuellen Maschine (`Ubuntu-20.04`) und Dateisysteme für Dockers Daten. Unter`Ubuntu-20.04`, Ihr Benutzerverzeichnis befindet sich unter`home\`*`username`*aus.

> **Note**
>
> Um von Ubuntu aus auf Dateien in Ihrer Windows-Installation zuzugreifen, navigieren Sie zu`/mnt/c`-Verzeichnis. Sie können beispielsweise Dateien in Ihrem Download-Verzeichnis auflisten, indem Sie`ls /mnt/c/Users/`*`windows-username`*`/Downloads`aus.

Mit Visual Studio Code können Sie Anwendungscode in Ihrer Entwicklungsumgebung bearbeiten und Befehle mit einem integrierten Terminal ausführen. Um Visual Studio Code zu installieren, besuchen Siecode.visualstudio.comaus. Fügen Sie nach der Installation dasRemote-SWSL-Erweiterung.

Das Windows-Terminal ist eine Alternative zum Standard-Ubuntu-Terminal, in dem Sie Befehle ausgeführt haben. Es unterstützt mehrere Registerkarten und kann PowerShell, Eingabeaufforderung und Terminals für jede andere Art von Linux ausführen, die Sie installieren. Es unterstützt das Kopieren und Einfügen mitCtrl+CundCtrl+V, anklickbare URLs und andere nützliche Verbesserungen. Um Windows Terminal zu installieren, besuchen Siemicrosoft.comaus.

Die AWS Panorama Panorama-API

Sie können die öffentliche API des AWS Panorama Panorama-Service verwenden, um Workflows für das Geräte- und Anwendungsmanagement zu automatisieren. Mit demAWS Command Line Interface oder demAWS SDK können Sie Skripts oder Anwendungen entwickeln, die Ressourcen und Bereitstellungen verwalten. Das GitHub Repository dieses Handbuchs umfasst Skripte, die Sie als Ausgangspunkt für Ihren eigenen Code verwenden können.

- aws-panorama-developer-guide/util-Skripte

Abschnitte

Automatisieren Sie die Geräteregistrierung

Um eine Appliance bereitzustellen, verwenden Sie dieProvisionDeviceAPI. Die Antwort enthält eine ZIP-Datei mit der Konfiguration des Geräts und den temporären Anmeldeinformationen. Dekodieren Sie die Datei und speichern Sie sie in einem Archiv mit dem Präfix`certificates-omni_`aus.

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}"
```

Die Anmeldeinformationen im Konfigurationsarchiv laufen nach 5 Minuten ab. Übertragen Sie das Archiv mit dem mitgelieferten USB-Laufwerk auf Ihre Appliance.

Um eine Kamera zu registrieren, benutzen Sie dieCreateNodeFromTemplateJobAPI. Diese API verwendet eine Zuordnung von Vorlagenparametern für den Benutzernamen, das Passwort und die URL der Kamera. Sie können diese Map mithilfe der Bash-Zeichenfolgenmanipulation als JSON-Dokument formatieren.

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)
```

Alternativ können Sie die JSON-Konfiguration auch aus einer Datei laden.

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

Appliances mit der AWS Panorama API verwalten

Mit der AWS Panorama Panorama-API können Sie Appliance-Management-Aufgaben automatisieren.

Geräte ansehen

Verwenden Sie die ListDevicesAPI, um eine Liste von Appliances mit Geräte-IDs abzurufen.

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

Verwenden Sie die DescribeDeviceAPI, um weitere Informationen zu einer Appliance zu erhalten.

```
$ 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
}
```

Appliance-Software aktualisieren

Wenn die`LatestSoftware` Version neuer ist als die`CurrentSoftware`, können Sie das Gerät aktualisieren. Verwenden Sie die CreateJobForDevicesAPI, um einen over-the-air (OTA-) Aktualisierungsjob zu erstellen.

```
$ 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"
        }
    ]
}
```

In einem Skript können Sie das Feld Bildversion in der Jobkonfigurationsdatei mit der Bearbeitung von Bash-Zeichenketten füllen.

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
}
```

Die Appliance lädt die angegebene Softwareversion herunter und aktualisiert sich selbst. Beobachten Sie den Fortschritt des Updates mit der 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
}
```

Um eine Liste aller laufenden Jobs zu erhalten, verwenden Sie die ListDevicesJobs.

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

Ein Beispielskript, das nach Updates sucht und diese anwendet, finden Sie in der Datei check-updates.sh im GitHub Repository dieses Handbuchs.

Appliances neu starten

Verwenden Sie die CreateJobForDevicesAPI, um eine Appliance neu zu starten.

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

```
            "JobId": "device-4tafxmplhtmzabv5lsacba4ere-0",
            "DeviceId": "device-4tafxmplhtmzabv5lsacba4ere"
        }
    ]
}
```

In einem Skript können Sie eine Liste von Geräten abrufen und eines auswählen, das interaktiv neu gestartet werden soll.

Example reboot-device.sh — Verwendung

```
$ ./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"
        }
    ]
}
```

Automatisieren Sie die Anwendungsbereitstellung

Um eine Anwendung bereitzustellen, verwenden Sie sowohl die AWS Panorama Application CLI als auchAWS Command Line Interface. Nachdem Sie den Anwendungscontainer erstellt haben, laden Sie ihn und andere Assets auf einen Amazon S3-Access Point hoch. Anschließend stellen Sie die Anwendung mit dem bereitCreateApplicationInstanceAPI.

Weitere Informationen und Anweisungen zur Verwendung der abgebildeten Skripts finden Sie in derBeispielanwendung README.

Abschnitte

- Baue den Container
- Laden Sie den Container hoch und registrieren Sie die Knoten
- Bereitstellen der Anwendung
- Überwachen Sie den Einsatz

Baue den Container

Um den Anwendungscontainer zu erstellen, verwenden Sie den`build-container`Befehl. Dieser Befehl erstellt einen Docker-Container und speichert ihn als komprimiertes Dateisystem im`assets`Ordner.

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
```

Sie können auch die Befehlszeilenvervollständigung verwenden, um das Pfadargument auszufüllen, indem Sie einen Teil des Pfads eingeben und dann drückenTAB.

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

Laden Sie den Container hoch und registrieren Sie die Knoten

Um die Bewerbung hochzuladen, verwenden Sie den`package-application`Befehl. Dieser Befehl lädt Assets aus dem hoch`assets`Ordner zu einem Amazon S3-Zugriffspunkt, den AWS Panorama verwaltet.

Example 4-package-app.sh

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

Die CLI der AWS Panorama-Anwendung lädt Container- und Deskriptor-Assets hoch, auf die in der Paketkonfiguration verwiesen wird (`package.json`) in jedem Paket und registriert die Pakete als Knoten in AWS Panorama. Anschließend verweisen Sie in Ihrem Anwendungsmanifest auf diese Knoten (`graph.json`), um die Anwendung bereitzustellen.

Bereitstellen der Anwendung

Um die Anwendung bereitzustellen, verwenden Sie denCreateApplicationInstanceAPI. Diese Aktion verwendet unter anderem die folgenden Parameter.

- `ManifestPayload`— Das Anwendungsmanifest (`graph.json`), das die Knoten, Pakete, Kanten und Parameter der Anwendung definiert.
- `ManifestOverridesPayload`— Ein zweites Manifest, das die Parameter im ersten Manifest außer Kraft setzt. Das Anwendungsmanifest kann als statische Ressource in der Anwendungsquelle betrachtet werden, wobei das Override-Manifest Einstellungen für die Bereitstellungszeit bereitstellt, mit denen die Bereitstellung angepasst werden kann.
- `DefaultRuntimeContextDevice`— Das Zielgerät.
- `RuntimeRoleArn`— Der ARN einer IAM-Rolle, die die Anwendung für den Zugriff auf AWS-Services und -Ressourcen verwendet.
- `ApplicationInstanceIdToReplace`— Die ID einer vorhandenen Anwendungsinstanz, die vom Gerät entfernt werden soll.

Die Payloads Manifest und Override sind JSON-Dokumente, die als Zeichenfolgenwert bereitgestellt werden müssen, der in einem anderen Dokument verschachtelt ist. Dazu lädt das Skript die Manifeste aus einer Datei als String und verwendet diejq-Toolum das verschachtelte Dokument zu erstellen.

Example 5-deploy.sh— Manifeste verfassen

```
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)"' <<< {})"
```

Das Deploy-Skript verwendet dieListDevicesAPI zum Abrufen einer Liste registrierter Geräte in der aktuellen Region und speichert die Auswahl des Benutzers in einer lokalen Datei für nachfolgende Bereitstellungen.

Example 5-deploy.sh— finde ein Gerät

```
    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)
```

Die Anwendungsrolle wird durch ein anderes Skript erstellt (1-create-role.sh). Das Deploy-Skript ruft den ARN dieser Rolle ab vonAWS CloudFormation. Wenn die Anwendung bereits auf dem Gerät bereitgestellt ist, ruft das Skript die ID dieser Anwendungsinstanz aus einer lokalen Datei ab.

Example 5-deploy.sh— Rollen-ARN und Ersatzargumente

```
# 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
```

Schließlich setzt das Skript alle Teile zusammen, um eine Anwendungsinstanz zu erstellen und die Anwendung auf dem Gerät bereitzustellen. Der Dienst antwortet mit einer Instanz-ID, die das Skript für die spätere Verwendung speichert.

Example 5-deploy.sh— Anwendung bereitstellen

```
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
```

Überwachen Sie den Einsatz

Um ein Deployment zu überwachen, verwenden Sie denListApplicationInstancesAPI. Das Monitor-Skript ruft die Geräte-ID und die Anwendungsinstanz-ID aus Dateien im Anwendungsverzeichnis ab und verwendet sie, um einen CLI-Befehl zu erstellen. Es ruft dann in einer Schleife auf.

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
```

Wenn die Bereitstellung abgeschlossen ist, können Sie die Protokolle einsehen, indem Sie Amazon anrufenCloudWatchProtokollierungs-API. Das View-Logs-Skript verwendet dieCloudWatchLogs`GetLogEvents`API.

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
```

Anwendungen mit der AWS Panorama API verwalten

Sie können Anwendungen mit der AWS Panorama Panorama-API überwachen und verwalten.

Anwendungen ansehen

Verwenden Sie die ListApplicationInstancesAPI, um eine Liste der Anwendungen abzurufen, die auf einer Appliance ausgeführt werden.

```
$ 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"
            }
        },
    ]
}
```

Verwenden Sie die ListApplicationInstanceNodeInstancesAPI, um weitere Informationen zu den Knoten einer Anwendungsinstanz zu erhalten.

```
$ 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"
        }
    ]
}
```

Kamerastreams verwalten

Mit der SignalApplicationInstanceNodeInstancesAPI können Sie Kamera-Stream-Knoten anhalten und wieder aufnehmen.

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

In einem Skript können Sie eine Liste von Knoten abrufen und einen auswählen, um interaktiv anzuhalten oder fortzufahren.

Example pause-camera.sh — Verwendung

```
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"
}
```

Wenn Sie die Kameraknoten anhalten und wieder aufnehmen, können Sie eine größere Anzahl von Kamerastreams durchlaufen, als gleichzeitig verarbeitet werden können. Ordnen Sie dazu mehrere Kamerastreams demselben Eingabeknoten in Ihrem Override-Manifest zu.

Im folgenden Beispiel ordnet das Override-Manifest zwei Kamerastreams`entrance-north` zu,`warehouse-floor` und zwar demselben Eingabeknoten (`camera_node`). Der`warehouse-floor` Stream ist aktiv, wenn die Anwendung gestartet wird und der`entrance-north` Knoten auf das Einschalten eines Signals wartet.

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"
                    }
                ]
            }
```

Einzelheiten zur Bereitstellung mit der API finden Sie unterAutomatisieren Sie die Anwendungsbereitstellung.

Verwenden eines VPC-Endpunkts

Wenn Sie in einer VPC ohne Internetzugang arbeiten, können Sie einen VPC-Endpunkt für die Verwendung mit AWS Panorama erstellen. Ein VPC-Endpunkt ermöglicht es Clients, die in einem privaten Subnetz laufen, sich ohne Internetverbindung mit einem AWS-Service zu verbinden.

Einzelheiten zu den Ports und Endpunkten, die von der AWS Panorama Appliance verwendet werden, finden Sie unter???.

Sections

Erstellung eines VPC-Endpunkts

Um eine private Verbindung zwischen Ihrer VPC und AWS Panorama herzustellen, erstellen Sie einen VPC-Endpunkt. Für die Verwendung von AWS Panorama ist kein VPC-Endpunkt erforderlich. Sie müssen nur dann einen VPC-Endpunkt erstellen, wenn Sie in einer VPC ohne Internetzugang arbeiten. Wenn die AWS-CLI oder das SDK versucht, eine Verbindung zu AWS Panorama herzustellen, wird der Datenverkehr über den VPC-Endpunkt geleitet.

Erstellen Sie einen VPC-Endpunkt für AWS Panorama mit den folgenden Einstellungen:

- Name des Dienstes — **`com.amazonaws.`*`us-west-2`*`.panorama`**
- Typ — Schnittstelle

Ein VPC-Endpunkt verwendet den DNS-Namen des Services, um ohne zusätzliche Konfiguration Traffic von AWS-SDK-Clients abzurufen. Weitere Informationen zur Verwendung von VPC-Endpunkten finden Sie unter Interface VPC Endpoints im Amazon VPC-Benutzerhandbuch.

Eine Appliance mit einem privaten Subnetz verbinden

Die AWS Panorama Appliance kann AWS über eine private VPN-Verbindung mit AWS Site-to-Site VPN oder eine Verbindung herstellenAWS Direct Connect. Mit diesen Services können Sie ein privates Subnetz erstellen, das sich bis zu Ihrem Rechenzentrum erstreckt. Die Appliance stellt eine Verbindung zum privaten Subnetz her und greift über VPC-Endpunkte auf AWS Dienste zu.

Site-to-Site VPN und AWS Direct Connect sind Dienste für die sichere Verbindung Ihres Rechenzentrums mit Amazon VPC. Mit Site-to-Site VPN können Sie handelsübliche Netzwerkgeräte verwenden, um eine Verbindung herzustellen. AWS Direct Connectverwendet ein AWS Gerät, um eine Verbindung herzustellen.

- Site-to-Site VPN — Was ist das? AWS Site-to-Site VPN
- AWS Direct Connect— Was ist? AWS Direct Connect

Nachdem Sie Ihr lokales Netzwerk mit einem privaten Subnetz in einer VPC verbunden haben, erstellen Sie VPC-Endpunkte für die folgenden Dienste.

- Amazon Simple Storage Service — AWS PrivateLinkfür Amazon S3
- AWS IoT Core— Verwendung AWS IoT Core mit VPC-Endpunkten mit Schnittstelle (Datenebene und Credential Provider)
- Amazon Elastic Container Registry — VPC-Endpunkte mit Amazon Elastic Container Registry-Schnittstelle
- Amazon CloudWatch — Verwendung von VPC-Endpunkten CloudWatch mit Schnittstelle
- Amazon CloudWatch Logs — Verwendung von CloudWatch Protokollen mit VPC-Endpunkten der Schnittstelle

Die Appliance benötigt keine Verbindung zum AWS Panorama Panorama-Service. Es kommuniziert mit AWS Panorama über einen Nachrichtenkanal inAWS IoT.

Zusätzlich zu VPC-Endpunkten AWS IoT erfordern Amazon S3 und Amazon die Verwendung von privaten gehosteten Zonen von Amazon Route 53. Die private gehostete Zone leitet den Datenverkehr von Subdomänen, einschließlich Subdomänen für Amazon S3 S3-Zugriffspunkte und MQTT-Themen, an den richtigen VPC-Endpunkt weiter. Informationen zu privat gehosteten Zonen finden Sie unter Arbeiten mit privat gehosteten Zonen im Amazon Route 53-Entwicklerhandbuch.

Eine VPC-Beispielkonfiguration mit VPC-Endpunkten und privaten gehosteten Zonen finden Sie unter. Beispielvorlagen AWS CloudFormation

Beispielvorlagen AWS CloudFormation

Das GitHub Repository für dieses Handbuch enthält AWS CloudFormation Vorlagen, mit denen Sie Ressourcen für die Verwendung mit AWS Panorama erstellen können. Die Vorlagen erstellen eine

VPC mit zwei privaten Subnetzen, einem öffentlichen Subnetz und einem VPC-Endpunkt. Sie können die privaten Subnetze in der VPC verwenden, um Ressourcen zu hosten, die vom Internet isoliert sind. Ressourcen im öffentlichen Subnetz können mit den privaten Ressourcen kommunizieren, aber auf die privaten Ressourcen kann nicht über das Internet zugegriffen werden.

Example vpc-endpoint.yml — Private Subnetze

```
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
  ...
```

Die `vpc-endpoint.yml` Vorlage zeigt, wie ein VPC-Endpunkt für AWS Panorama erstellt wird. Sie können diesen Endpunkt verwenden, um AWS-Panorama-Ressourcen mit dem AWS SDK oder zu verwaltenAWS CLI.

Example vpc-endpoint.yml — VPC-Endpunkt

```
  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:
            - "*"
```

Das `PolicyDocument` ist eine ressourcenbasierte Berechtigungsrichtlinie, die die API-Aufrufe definiert, die mit dem Endpunkt getätigt werden können. Sie können die Richtlinie ändern, um die Aktionen und Ressourcen einzuschränken, auf die über den Endpunkt zugegriffen werden kann. Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon-VPC-Benutzerhandbuch.

Die `vpc-appliance.yml` Vorlage zeigt, wie VPC-Endpunkte und private Hosting-Zonen für Services erstellt werden, die von der AWS Panorama Appliance verwendet werden.

Example vpc-appliance.yml — Amazon S3 S3-Zugriffspunkt-Endpunkt mit privat gehosteter Zone

```
  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 ] ] ]
```

Die Beispielvorlagen demonstrieren die Erstellung von Amazon VPC- und Route 53-Ressourcen mit einer Beispiel-VPC. Sie können diese an Ihren Anwendungsfall anpassen, indem Sie die VPC-Ressourcen entfernen und die Verweise auf Subnetz-, Sicherheitsgruppen- und VPC-IDs durch die IDs Ihrer Ressourcen ersetzen.

Beispielanwendungen, Skripte und Vorlagen

Das GitHub Das Repository für dieses Handbuch enthält Beispielanwendungen, Skripte und Vorlagen fürAWS PanoramaGeräte. Verwenden Sie diese Beispiele, um bewährte Verfahren zu erlernen und Entwicklungsabläufe zu automatisieren.

Abschnitte

Beispielanwendungen

Beispielanwendungen demonstrieren die Verwendung vonAWS PanoramaFunktionen und allgemeine Computer-Vision-Aufgaben. Zu diesen Beispielanwendungen gehören Skripts und Vorlagen, die die Einrichtung und Bereitstellung automatisieren. Mit minimaler Konfiguration können Sie Anwendungen von der Befehlszeile aus bereitstellen und aktualisieren.

- aws-panorama-sample— Grundlegendes maschinelles Sehen mit einem Klassifikationsmodell. Verwenden Sie dieAWS SDK for Python (Boto)um Metriken hochzuladen auf CloudWatch, Instrumentenvorverarbeitungs- und Inferenzmethoden und Konfiguration der Protokollierung.
- Debug-Server—Eingehende Ports öffnenauf dem Gerät und leiten Sie den Datenverkehr an einen Anwendungscode-Container weiter. Verwenden Sie Multithreading, um Anwendungscode, einen HTTP-Server und einen HTTP-Client gleichzeitig auszuführen.
- benutzerdefiniertes Modell— Modelle aus dem Code exportieren und mit kompilieren SageMaker Neo, um die Kompatibilität mit dem zu testenAWS PanoramaGerät. Erstellen Sie lokal in einer Python-Entwicklung, in einem Docker-Container oder auf einer Amazon EC2-Instance. Exportieren und kompilieren Sie alle integrierten Anwendungsmodelle in Keras für ein bestimmtes TensorFlow oder Python-Version.

Weitere Beispielanwendungen finden Sie auch aufaws-panorama-samplesRepositorium.

Dienstprogramm-Skripte

Die Skripte in der`util-scripts`Verzeichnis verwaltenAWS PanoramaRessourcen oder automatisieren Sie Entwicklungsworkflows.

- provision-device.sh— Stellen Sie ein Gerät bereit.
- check-updates.sh— Suchen Sie nach Softwareupdates für die Appliance und wenden Sie sie an.
- reboot-device.sh— Starte ein Gerät neu.
- register-camera.sh— Registriere eine Kamera.
- deregister-camera.sh— Löscht einen Kameraknoten.
- view-logs.sh— Logs für eine Anwendungsinstanz anzeigen.
- pause-camera.sh— Einen Kamerastream anhalten oder fortsetzen.
- push.sh— Eine Anwendung erstellen, hochladen und bereitstellen.
- rename-package.sh— Benennt ein Node-Paket um. Aktualisiert Verzeichnisnamen, Konfigurationsdateien und das Anwendungsmanifest.
- samplify.sh— Ersetzen Sie Ihre Konto-ID durch eine Beispielkonto-ID und stellen Sie Backup-Konfigurationen wieder her, um die lokale Konfiguration zu entfernen.
- update-model-config.sh— Fügen Sie das Modell nach der Aktualisierung der Deskriptordatei erneut zur Anwendung hinzu.
- cleanup-patches.sh— Alte Patch-Versionen abmelden und ihre Manifeste aus Amazon S3 löschen.

Einzelheiten zur Nutzung finden Sie unterdie README-Datei.

AWS CloudFormation-Vorlagen

Benutze dieAWS CloudFormationVorlagen in der`cloudformation-templates`Verzeichnis, für das Ressourcen erstellt werden sollenAWS PanoramaAnwendungen.

- alarm-application.yml— Erstellen Sie einen Alarm, der eine Anwendung auf Fehler überwacht. Wenn die Anwendungsinstanz Fehler ausgibt oder für 5 Minuten nicht mehr läuft, sendet der Alarm eine Benachrichtigungs-E-Mail.

- alarm-device.yml— Erstellen Sie einen Alarm, der die Konnektivität eines Geräts überwacht. Wenn das Gerät für 5 Minuten keine Messwerte mehr sendet, sendet der Alarm eine Benachrichtigungs-E-Mail.
- application-role.yml— Erstellen Sie eine Anwendungsrolle. Die Rolle beinhaltet die Erlaubnis, Metriken zu senden an CloudWatch. Fügen Sie der Richtlinienerklärung Berechtigungen für andere API-Operationen hinzu, die Ihre Anwendung verwendet.
- vpc-appliance.yml— Erstellen Sie eine VPC mit privatem Subnetzdienstzugriff fürAWS PanoramaGerät. Um die Appliance mit einer VPC zu verbinden, verwenden SieAWS Direct ConnectoderAWS Site-to-Site VPN.
- vpc-endpoint.yml— Erstellen Sie eine VPC mit privatem Subnetzdienstzugriff aufAWS PanoramaDienst. Ressourcen innerhalb der VPC können eine Verbindung herstellen zuAWS Panoramazu überwachen und zu verwaltenAWS PanoramaRessourcen ohne Verbindung zum Internet.

Die`create-stack.sh`Das Skript in diesem Verzeichnis erstelltAWS CloudFormationStapel. Es benötigt eine variable Anzahl von Argumenten. Das erste Argument ist der Name der Vorlage, und die übrigen Argumente sind Überschreibungen für Parameter in der Vorlage.

Mit dem folgenden Befehl wird beispielsweise eine Anwendungsrolle erstellt.

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

Weitere Beispiele und Tools

Dasaws-panorama-samplesDas Repository enthält mehr Beispielanwendungen und nützliche Tools.

- Anwendungen— Beispielanwendungen für verschiedene Modellarchitekturen und Anwendungsfälle.
- Validierung des Kamerastreams— Validieren Sie Kamerastreams.
- PanoJupyter— Ausführen JupyterLab auf einemAWS PanoramaGerät.
- Seitenladen— Aktualisieren Sie den Anwendungscode, ohne einen Anwendungscontainer zu erstellen oder bereitzustellen.

DerAWSDie Community hat auch Tools und Anleitungen entwickelt fürAWS Panorama. Schauen Sie sich die folgenden Open-Source-Projekte an GitHub.

- Ausstechformer-Panorama— Eine Cookiecutter-Vorlage fürAWS PanoramaAnwendungen.
- Rucksack— Python-Module für den Zugriff auf Details zur Laufzeitumgebung, Profilerstellung und zusätzliche Videoausgabeoptionen.

ÜberwachungAWS Panoramaressourcen und -anwendungen

Sie können überwachenAWS PanoramaRessourcen imAWS Panoramakonsole und mit AmazonCloudWatchaus. DieAWS PanoramaAppliance stellt eine Verbindung zumAWSCloud über das Internet, um seinen Status und den Status verbundener Kameras zu melden. Während es aktiviert ist, sendet die Appliance auch Protokolle anCloudWatchMeldet sich in Echtzeit an.

Die Appliance erhält die Berechtigung zur VerwendungAWS IoT,CloudWatchProtokolle und andere AWS-Services von einer Servicerolle, die Sie bei der ersten Verwendung desAWS Panoramaconsole. Weitere Informationen finden Sie unter AWS Panorama Panorama-Servicerollen und serviceübergreifende Ressourcen .

Hilfe bei der Behebung bestimmter Fehler finden Sie unterFehlerbehebungaus.

Themen

- Überwachung in der AWS Panorama Panorama-Konsole
- Anzeigen von AWS Panorama Panorama-Proto
- Überwachen von Appliances und Anwendungen mit AmazonCloudWatch

Überwachung in der AWS Panorama Panorama-Konsole

Sie können die AWS Panorama-Konsole verwenden, um Ihre AWS Panorama Appliance und Kameras zu überwachen. Die Konsole verwendetAWS IoTum den Status der Appliance zu überwachen.

So überwachen Sie Ihre Appliance in der AWS Panorama Panorama-Konsole

1. Öffnen SieAWS Panorama -Konsoleaus.
2. Öffnen Sie die AWS Panorama -KonsoleSeite „Geräte“aus.
3. Wählen Sie ein Gerät aus.
4. Um den Status einer Anwendungsinstanz anzuzeigen, wählen Sie sie aus der Liste aus.
5. Um den Status der Netzwerkschnittstellen der Appliance anzuzeigen, wählen SieEinstellungenaus.

Der Gesamtstatus der Appliance wird oben auf der Seite angezeigt. Wenn der Status lautetStatus "Online", dann ist die Appliance mit verbundenAWSund regelmäßige Statusaktualisierungen senden.

Anzeigen von AWS Panorama Panorama-Proto

AWS Panorama meldet Anwendungs- und Systemereignisse an Amazon CloudWatch Logs. Wenn Sie auf Probleme stoßen, können Sie die Ereignisprotokolle verwenden, um Ihre AWS Panorama Panorama-Anwendung zu debuggen oder Fehler bei der Konfiguration der Anwendung zu beheben.

Anzeigen von Protokollen CloudWatch Logs (Protokolle)

1. Öffne denSeite „Logs-Gruppen“ der CloudWatch -Protokoll-Konsole.
2. Suchen Sie AWS Panorama Anwendungs- und Appliance-Protokolle in den folgenden Gruppen:

   - Geräteprotokolle–`/aws/panorama/devices/device-id`
   - Anwendungs-Logs–`/aws/panorama/devices/device-id/applications/instance-id`

Wenn Sie eine Appliance nach dem Update der Systemsoftware erneut bereitstellen, können Sie auchAnzeigen von Protokollen auf dem Bereitstellungs-USB-Laufwerk.

Abschnitte

Anzeigen von Geräteprotokoll

Die AWS Panorama Appliance erstellt eine Protokollgruppe für das Gerät und eine Gruppe für jede Anwendungsinstanz, die Sie bereitstellen. Die Geräteprotokolle enthalten Informationen über den Anwendungsstatus, Software-Upgrades und Systemkonfiguration.

Geräteprotokolle —**`/aws/panorama/devices/device-id`**

- `occ_log`— Ausgabe aus dem Controller-Prozess. Dieser Prozess koordiniert Anwendungsbereitstellungen und berichtet über den Status der Knoten jeder Anwendungsinstanz.
- `ota_log`— Ausgabe aus dem Prozess, der koordiniert over-the-air (OTA) Softwareupgrades.

- `syslog`— Ausgabe aus dem Syslog-Prozess des Geräts, der zwischen Prozessen gesendete Nachrichten erfasst.
- `kern_log`— Ereignisse aus dem Linux-Kernel des Geräts.
- `logging_setup_logs`— Ausgabe aus dem Prozess, der die CloudWatch Logs-Agenten.
- `cloudwatch_agent_logs`— Ausgabe des CloudWatch Logs-Agenten.
- `shadow_log`— Ausgabe desAWS IoTDevice Shadow.

Anzeigen von Anwendungsprotokoll

Die Protokollgruppe einer Anwendungsinstanz enthält einen Protokollstrom für jeden Knoten, der nach dem Knoten benannt ist.

Anwendungsprotokolle —**`/aws/panorama/devices/`*`device-id`*`/applications/`*`instance-id`***

- Code— Ausgabe aus Ihrem Anwendungscode und dem AWS Panorama Application SDK. Aggregiert Anwendungsprotokolle von`/opt/aws/panorama/logs`.
- Model— Ausgabe aus dem Prozess, der Inferenzanforderungen mit einem Modell koordiniert.
- Stream— Ausgabe des Prozesses, der Videos aus einem Kamerastream dekodiert.
- Anzeigen— Ausgabe aus dem Prozess, der die Videoausgabe für den HDMI-Anschluss rendert.
- `mds`— Protokolle vom Metadatenserver der Appliance.
- `console_output`— Errors Standard Output- und Errors -Streams von

Wenn dies nicht der Fall ist, werden Sie angemeldet CloudWatch Protokollen bestätigen, dass Sie sich in der korrekten AWS-Region befinden. Falls dies der Fall ist, könnte es ein Problem mit der Verbindung der Appliance zu AWS oder mit den Berechtigungen fürdie ApplianceAWS Identity and Access Management(IAM) -Rolle.

Konfigurieren von Anwendungsprotokoll

Konfigurieren eines Python-Loggers zum Schreiben von Protokolldateien`/opt/aws/panorama/logs`. Die Appliance streamt Protokolle von diesem Speicherort an CloudWatch Logs. Um zu vermeiden, dass zu viel Speicherplatz verwendet wird, verwenden Sie eine maximale Dateigröße von 10 MiB und eine Sicherungsanzahl von 1. Im folgenden Beispiel wird eine Methode gezeigt, die einen Logger erstellt.

Example application.py— Logger-Konfiguration

```
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
```

Initialisieren Sie den Logger im globalen Bereich und verwenden Sie ihn im gesamten Anwendungscode.

Example application.py— Initialize Logger

```
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()
```

Anzeigen von Provisioningproto

Während der Bereitstellung kopiert die AWS Panorama Appliance Protokolle auf das USB-Laufwerk, das Sie zum Übertragen des Konfigurationsarchivs auf die Appliance verwenden. Verwenden Sie diese Protokolle, um Bereitstellungsprobleme auf Appliances mit der neuesten Softwareversion zu beheben.

> **Important**
>
> Bereitstellungsprotokolle sind für Appliances verfügbar, die auf Softwareversion 4.3.23 oder höher aktualisiert wurden.

Anwendungsprotokolle

- `/panorama/occ.log`— Softwareprotokolle für AWS Panorama Panorama-Controller.
- `/panorama/ota_agent.log`— AWS Panorama over-the-air Agentenprotokolle aktualisieren.
- `/panorama/syslog.log`— Linux-Systemprotokolle.
- `/panorama/kern.log`— Linux-Kernel-Protokolle.

Ausgeben von Protokollen von einem Gerät

Wenn Ihre Geräte- und Anwendungsprotokolle nicht in angezeigt werden CloudWatch Protokolle, Sie können ein USB-Laufwerk verwenden, um ein verschlüsseltes Protokollabbild vom Gerät abzurufen. Das AWS Panorama Panorama-Serviceteam kann die Protokolle in Ihrem Namen entschlüsseln und Sie beim Debuggen unterstützen.

Voraussetzungen

Um das Verfahren zu befolgen, benötigen Sie die folgende Hardware:

- USB-Laufwerk— Ein FAT32-formatiertes USB-Flash-Speicherlaufwerk mit mindestens 1 GB Speicherplatz zum Übertragen der Protokolldateien von der AWS Panorama Appliance.

So geben Sie Protokolle vom Gerät aus

1. Bereiten Sie ein USB-Laufwerk mit einem`managed_logs`Ordner in einem`panorama`-Ordners.

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

2. Connect Sie das USB-Laufwerk mit dem Gerät.
3. Ausschaltendie AWS Panorama Appliance.

4. Schalten Sie die AWS Panorama Appliance ein.
5. Das Gerät kopiert Protokolle auf das Gerät. Die Status-LEDblinkt blauwährend dies in Bearbeitung ist.
6. Protokolldateien können dann darin gefunden werdenmanaged_logsVerzeichnis mit dem Formatpanorama_device_log_*v1_dd_hh_mm*.img

Sie können das Logbild nicht selbst entschlüsseln. Arbeiten Sie mit dem Kundensupport, einem technischen Kundenbetreuer für AWS Panorama oder einem Lösungsarchitekten zusammen, um sich mit dem Serviceteam abzustimmen.

Überwachen von Appliances und Anwendungen mit AmazonCloudWatch

Wenn eine Appliance online ist, sendet AWS Panorama Metriken an AmazonCloudWatchaus. Sie können Graphen und Dashboards mit diesen Metriken imCloudWatch-Konsole zur Überwachung der Appliance-Aktivitäten und zum Festlegen von Alarmen, die Sie benachrichtigen, wenn Geräte offline gehen oder Anwendungen auf Fehler stoßen.

So zeigen Sie Metriken in der CloudWatch-Konsole an

1. Öffnen SieSeite „Metriken der AWS Panorama Panorama-Konsole“(`PanoramaDeviceMetrics`-Namespace).
2. Wählen Sie ein Dimensionsschema aus.
3. Wählen Sie Metriken aus, um sie dem Diagramm hinzuzufügen.
4. Um eine andere Statistik auszuwählen und das Diagramm anzupassen, verwenden Sie die Optionen auf der Registerkarte Graphed metrics (Graphierte Metriken). Standardmäßig verwenden Diagramme die `Average`-Statistik für alle Metriken.

> **Preise**
>
> CloudWatchhat ein Always Free Kontingent. Über den Schwellenwert für das kostenlose Kontingent hinaus berechnet CloudWatch Gebühren für Metriken, Dashboards, Alarme, Protokolle und Insights. Weitere Details finden Sie unter CloudWatch-Preise.

Weitere Informationen zuCloudWatch, finden Sie imAmazonCloudWatch-Benutzerhandbuchaus.

Abschnitte

- Verwenden von Gerätemetriken
- Verwenden von Anwendungsmetriken
- Konfigurieren von Alarmen

Verwenden von Gerätemetriken

sendet bei Online-Metriken an AmazonCloudWatchaus. Sie können diese Metriken verwenden, um die Geräteaktivität zu überwachen und einen Alarm auszulösen, wenn Geräte offline gehen.

- `DeviceActive`— Wird regelmäßig gesendet, wenn das Gerät aktiv ist.

  Maße —`DeviceId`und`DeviceName`aus.

Aufrufen des`DeviceActive`-Metrik bei`Average`Statistik.

Verwenden von Anwendungsmetriken

sendet bei einer Anwendung Metriken Metriken an AmazonCloudWatchaus. Sie können diese Metriken verwenden, um einen Alarm auszulösen, wenn eine Anwendung nicht mehr ausgeführt wird.

- `ApplicationErrors`— Die Anzahl der aufgezeichneten Anwendungsfehler.

  Maße —`ApplicationInstanceName`und`ApplicationInstanceId`aus.

Zeigen Sie die Anwendungsmetriken mit`Sum`Statistik.

Konfigurieren von Alarmen

Um Benachrichtigungen zu erhalten, wenn eine Metrik einen Schwellenwert überschreitet, erstellen Sie einen Alarm. Sie können z. B. einen Alarm erstellen, der eine Benachrichtigung sendet, wenn die Summe der`ApplicationErrors`Die Metrik bleibt 20 Minuten lang bei 1.

So erstellen Sie einen Alarm

1. Öffnen SieAmazonCloudWatchSeite Alarme der Konsoleaus.
2. Wählen Sie Create Alarm aus.
3. Klicken Sie aufAuswählen von Metrikund suchen Sie eine Metrik für Ihr Gerät, z.`ApplicationErrors`zum`applicationInstance-gk75xmplqbqtenlnmz4ehiu7xa`,`my-application`aus.
4. Folgen Sie den Anweisungen, um eine Bedingung, eine Aktion und einen Namen für den Alarm zu konfigurieren.

Detaillierte Anweisungen finden Sie unter .Erstellen einesCloudWatchAlarmimAmazonCloudWatch-Benutzerhandbuchaus.

Fehlerbehebung

Die folgenden Themen enthalten Hinweise zur Fehlerbehebung bei Fehlern und Problemen, die bei der Verwendung der AWS Panorama Konsole, Appliance oder des SDK auftreten können. Wenn Sie ein Problem finden, das hier nicht aufgeführt ist, verwenden Sie die Schaltfläche Feedback geben auf dieser Seite, um es zu melden.

Sie finden die Protokolle für Ihre Appliance in der Amazon CloudWatch Logs-Konsole. Die Appliance lädt Protokolle aus Ihrem Anwendungscode und der Appliance-Software hoch und AWS IoT verarbeitet sie, sobald sie generiert werden. Weitere Informationen finden Sie unter Anzeigen von AWS Panorama Panorama-Proto.

Bereitstellung

Problem: (macOS) Mein Computer erkennt das mitgelieferte USB-Laufwerk mit einem USB-C-Adapter nicht.

Dies kann auftreten, wenn Sie das USB-Laufwerk an einen USB-C-Adapter anschließen, der bereits an Ihren Computer angeschlossen ist. Versuchen Sie, den Adapter zu trennen und ihn erneut anzuschließen, während das USB-Laufwerk bereits angeschlossen ist.

Problem: Die Bereitstellung schlägt fehl, wenn ich mein eigenes USB-Laufwerk verwende.

Problem: Die Bereitstellung schlägt fehl, wenn ich den USB 2.0-Anschluss der Appliance verwende.

Die AWS Panorama Appliance ist mit USB-Flash-Speichergeräten zwischen 1 und 32 GB kompatibel, aber nicht alle sind kompatibel. Bei der Verwendung des USB 2.0-Anschlusses für die Bereitstellung wurden einige Probleme beobachtet. Verwenden Sie für konsistente Ergebnisse das mitgelieferte USB-Laufwerk mit dem USB 3.0-Anschluss (neben dem HDMI-Anschluss).

Beim Lenovo ThinkEdge® SE70 ist kein USB-Laufwerk im Lieferumfang des Geräts enthalten. Verwenden Sie ein USB 3.0-Laufwerk mit mindestens 1 GB Speicher.

Konfiguration der Appliance

Problem: Die Appliance zeigt beim Booten einen leeren Bildschirm an.

Nach Abschluss der ersten Startsequenz, die etwa eine Minute dauert, zeigt die Appliance mindestens eine Minute lang einen leeren Bildschirm an, während Ihr Modell geladen und Ihre

Anwendung gestartet wird. Außerdem gibt die Appliance kein Video aus, wenn Sie nach dem Einschalten ein Display anschließen.

Problem: Das Gerät reagiert nicht, wenn ich den Netzschalter gedrückt halte, um es auszuschalten.

Das sichere Herunterfahren des Geräts dauert bis zu 10 Sekunden. Sie müssen den Netzschalter nur 1 Sekunde lang gedrückt halten, um die Abschaltsequenz zu starten. Eine vollständige Liste der Tastenoperationen finden Sie unterTasten und Lichter der AWS Panorama Appliance.

Problem: Ich muss ein neues Konfigurationsarchiv generieren, um Einstellungen zu ändern oder ein verloren gegangenes Zertifikat zu ersetzen.

AWS Panoramaspeichert das Gerätezertifikat oder die Netzwerkkonfiguration nicht, nachdem Sie es heruntergeladen haben, und Sie können Konfigurationsarchive nicht wiederverwenden. Löschen Sie die Appliance mithilfe der AWS Panorama Konsole und erstellen Sie eine neue Appliance mit einem neuen Konfigurationsarchiv.

Anwendungskonfiguration

Problem: Wenn ich mehrere Anwendungen starte, kann ich nicht kontrollieren, welche den HDMI-Ausgang verwenden.

Wenn Sie mehrere Anwendungen mit Ausgangsknoten bereitstellen, verwendet die zuletzt gestartete Anwendung den HDMI-Ausgang. Wenn diese Anwendung nicht mehr ausgeführt wird, kann eine andere Anwendung die Ausgabe verwenden. Um nur einer Anwendung Zugriff auf die Ausgabe zu gewähren, entfernen Sie den Ausgabeknoten und den entsprechenden Edge aus dem Anwendungsmanifest der anderen Anwendung und stellen Sie sie erneut bereit.

Problem: Die Anwendungsausgabe erscheint nicht in den Protokollen

Konfigurieren Sie einen Python-Logger, in den Protokolldateien geschrieben werden sollen`/opt/aws/panorama/logs`. Diese werden in einem Protokollstream für den Code-Container-Knoten erfasst. Standardausgabe- und Fehlerdatenströme werden in einem separaten Protokollstream namens erfasst`console-output`. Wenn Sie dies verwenden`print`, verwenden Sie die `flush=True` Option, um zu verhindern, dass Nachrichten im Ausgabepuffer hängen bleiben.

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

Quelle: AWS Panorama Service

Jedes Mal, wenn Sie eine Änderung an einer Anwendung implementieren, registrieren Sie eine Patch-Version, die die Paketkonfiguration und die Asset-Dateien für jedes verwendete Paket darstellt. Verwenden Sie das Cleanup-Patches-Skript, um ungenutzte Patch-Versionen zu deregistrieren.

Kamerastreams

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

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

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

Quelle: Kameraknotenprotokoll

Die Appliance kann keine Verbindung zum Kamerastream der Anwendung herstellen. In diesem Fall ist die Videoausgabe leer oder friert beim zuletzt verarbeiteten Frame ein, während die Anwendung auf ein Videoframe vom AWS Panorama Anwendungs-SDK wartet. Die Appliance-Software versucht, eine Verbindung zum Kamerastream herzustellen, und protokolliert Timeout-Fehler im Kameraknotenprotokoll. Stellen Sie sicher, dass Ihre Kamerastream-URL korrekt ist und dass der RTSP-Verkehr zwischen der Kamera und der Appliance innerhalb Ihres Netzwerks routingfähig ist. Weitere Informationen finden Sie unter Die AWS Panorama Appliance mit Ihrem Netzwerk verbinden.

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

Quelle: OCC-Protokoll

Das AWS Secrets Manager Geheimnis mit den Anmeldeinformationen des Kamerastreams konnte nicht gefunden werden. Löschen Sie den Kamerastream und erstellen Sie ihn neu.

Fehler: Camera did not provide an H264 encoded stream

Quelle: Kameraknotenprotokoll

Der Kamerastream hat eine andere Kodierung als H.264, z. B. H.265. Stellen Sie die Anwendung erneut mit einem H.264-Kamerastream bereit. Einzelheiten zu unterstützten Kameras finden Sie unter. Unterstützte Kameras

Sicherheit in AWS Panorama

Die Sicherheit in der Cloud hat AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die zur Erfüllung der Anforderungen von Organisationen entwickelt wurden, für die Sicherheit eine kritische Bedeutung hat.

Sicherheit gilt zwischen AWS und Ihnen eine geteilte Verantwortung. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS-Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für AWS Panorama gelten, finden Sie unterAWS-Dienstleistungen in Scope nach Compliance-Programmaus.
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation veranschaulichen, wie das Modell der geteilten Verantwortung bei der Verwendung von AWS Panorama angewendet wird. Die folgenden Themen veranschaulichen, wie Sie AWS Panorama zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren auch, wie Sie andere AWS-Services nutzen können, die Ihnen helfen, Ihre AWS Panorama -Ressourcen zu überwachen und zu sichern.

Themen

Sicherheitsfunktionen der AWS Panorama Appliance

Schutz IhrerAnwendungen, Modelleund Hardware gegen bösartigen Code und andere Exploits implementiert die AWS Panorama Appliance umfangreiche Sicherheitsfunktionen. Dazu gehört u. a. Folgendes:

- Vollständige Festplattenverschlüsselung— Die Appliance implementiert Vollfestplattenverschlüsselung (LUKS2) mit Linux Unified Key Setup. Alle Systemsoftware- und Anwendungsdaten werden mit einem für Ihr Gerät spezifischen Schlüssel verschlüsselt. Selbst bei physischem Zugriff auf das Gerät kann ein Angreifer den Inhalt seines Speichers nicht überprüfen.
- Randomisierung des Speicher-Layouts— Zum Schutz vor Angriffen, die auf ausführbaren Code abzielen, der in den Speicher geladen wurde, verwendet die AWS Panorama Appliance Randomisierung (ASLR). ASLR randomisiert den Speicherort des Betriebssystemcodes, während er in den Speicher geladen wird. Dies verhindert die Verwendung von Exploits, die versuchen, bestimmte Codeabschnitte zu überschreiben oder auszuführen, indem vorausgesagt wird, wo sie zur Laufzeit gespeichert sind.
- Vertraute Ausführungsumgebung— Die Appliance verwendet eine vertrauenswürdige Ausführungsumgebung (TEE) basierend auf ARM TrustZone mit isoliertem Speicher-, Speicher- und Verarbeitungsressourcen. Schlüssel und andere sensible Daten, die in der Vertrauenszone gespeichert sind, kann nur von einer vertrauenswürdigen Anwendung zugegriffen werden, die in einem separaten Betriebssystem innerhalb des TEE ausgeführt wird. Die AWS Panorama Appliance-Software läuft zusammen mit Anwendungscode in der nicht vertrauenswürdigen Linux-Umgebung. Es kann nur auf kryptografische Operationen zugreifen, indem es eine Anfrage an die sichere Anwendung stellt.
- Sichere Bereitstellung— Wenn Sie eine Appliance bereitstellen, sind die Anmeldeinformationen (Schlüssel, Zertifikate und anderes kryptografisches Material), die Sie auf das Gerät übertragen, nur für kurze Zeit gültig. Die Appliance verwendet die kurzlebigen Anmeldeinformationen, um eine Verbindung herzustellenAWS IoTund fordert ein Zertifikat für sich selbst an, das länger gültig ist. Der AWS Panorama Panorama-Dienst generiert Anmeldeinformationen und verschlüsselt sie mit einem Schlüssel, der auf dem Gerät fest codiert ist. Nur das Gerät, das das Zertifikat angefordert hat, kann es entschlüsseln und mit AWS Panorama kommunizieren.
- Sicheres Starten— Beim Start des Geräts wird jede Softwarekomponente authentifiziert, bevor sie ausgeführt wird. Das Boot-ROM, Software, die im Prozessor fest codiert ist und nicht geändert werden kann, entschlüsselt den Bootloader mit einem fest codierten Verschlüsselungsschlüssel, der den Kernel der vertrauenswürdigen Ausführungsumgebung validiert, usw.

- Signierter Kernel— Kernel-Module sind mit einem asymmetrischen Verschlüsselungsschlüssel signiert. Der Betriebssystemkernel entschlüsselt die Signatur mit dem öffentlichen Schlüssel und überprüft, ob sie mit der Signatur des Moduls übereinstimmt, bevor das Modul in den Speicher geladen wird.
- dm-verity— Ähnlich wie Kernelmodule validiert werden, verwendet die Appliance die Linux Device Mappersdm-verity-Funktion, um die Integrität des Software-Images vor dem Einhängen zu überprüfen. Wenn die Appliance-Software geändert wird, wird sie nicht ausgeführt.
- Rollback-Prävention— Wenn Sie die Appliance-Software aktualisieren, bläst das Gerät eine elektronische Sicherung auf den SoC (System auf einem Chip). Jede Softwareversion geht davon aus, dass eine zunehmende Anzahl von Sicherungen durchgebrannt wird, und kann nicht ausgeführt werden, wenn mehr durchgebrannt sind.

Bewährte Methoden für Sicherheitsmaßnahmen für AWS Panorama

Beachten Sie bei der Verwendung der AWS Panorama Panorama-Appliance die folgenden Best Practices.

- Sichern Sie das Gerät physisch— Installieren Sie die Appliance in einem geschlossenen Server-Rack oder einem sicheren Raum. Beschränken Sie den physischen Zugriff auf das Gerät auf autorisiertes Personal.
- Sichern Sie die Netzwerkverbindung der Appliance— Connect Sie die Appliance mit einem Router, der den Zugriff auf interne und externe Ressourcen einschränkt. Die Appliance muss eine Verbindung zu Kameras herstellen, die sich in einem sicheren internen Netzwerk befinden können. Es muss auch eine Verbindung herstellenAWSaus. Verwenden Sie den zweiten Ethernet-Port nur für physische Redundanz und konfigurieren Sie den Router so, dass er nur erforderlichen Datenverkehr zulässt.

  Verwenden Sie eine der empfohlenen Netzwerkkonfigurationen, um Ihr Netzwerk-Layout zu planen. Weitere Informationen finden Sie unter Die AWS Panorama Appliance mit Ihrem Netzwerk verbinden.
- Formatieren des USB-Laufwerks— Entfernen Sie nach der Bereitstellung einer Appliance das USB-Laufwerk und formatieren Sie es. Die Appliance verwendet das USB-Laufwerk nicht, nachdem es sich beim AWS Panorama Panorama-Dienst registriert hat. Formatieren Sie das Laufwerk, um temporäre Anmeldeinformationen, Konfigurationsdateien und Bereitstellungsprotokolle zu entfernen.
- Halten Sie das Gerät auf dem neuesten Stand halten— Wenden Sie Softwareupdates der Appliance zeitnah an. Wenn Sie eine Appliance in der AWS Panorama Panorama-Konsole anzeigen, benachrichtigt Sie die Konsole, wenn ein Softwareupdate verfügbar ist. Weitere Informationen finden Sie unter Verwaltung einer AWS Panorama Panorama-Appliance.

  Mit derDescribeDeviceAPI-Betrieb können Sie die Überprüfung nach Updates automatisieren, indem Sie die`LatestSoftware`und`CurrentSoftware`unterscheiden sich nicht. Wenn sich die neueste Softwareversion von der aktuellen Version unterscheidet, wenden Sie das Update mit der Konsole an oder verwenden Sie dieCreateJobForDevicesverwenden.
- Wenn Sie eine Appliance nicht mehr verwenden, setzen Sie sie zurück- Bevor Sie die Appliance aus Ihrem sicheren Rechenzentrum verschieben, setzen Sie sie vollständig zurück. Drücken Sie bei ausgeschaltetem und eingestecktem Gerät 5 Sekunden lang sowohl die Power- als auch

die Reset-Taste gleichzeitig. Dadurch werden Kontoanmeldeinformationen, Anwendungen und Protokolle von der Appliance gelöscht.

Weitere Informationen finden Sie unter Tasten und Lichter der AWS Panorama Appliance.

- Beschränken Sie den Zugriff auf AWS Panorama und andere AWS-Services— DieAWSPanoramaFullAccessbietet Zugriff auf alle AWS Panorama API-Vorgänge und gegebenenfalls Zugriff auf andere Dienste. Wenn möglich, schränkt die Richtlinie den Zugriff auf Ressourcen auf der Grundlage von Benennungskonventionen ein. Zum Beispiel bietet es Zugriff aufAWS Secrets ManagerGeheimnisse, die Namen haben, die mit`panorama`aus. Verwenden Sie für Benutzer, die schreibgeschützten Zugriff oder Zugriff auf einen spezifischeren Satz von Ressourcen benötigen, die verwaltete Richtlinie als Ausgangspunkt für Ihre Richtlinien mit den geringsten Berechtigungen.

  Weitere Informationen finden Sie unter Identitätsbasierte IAM-Richtlinien für AWS Panorama.

Datenschutz in AWS Panorama

Das Modell der AWS geteilten gilt für den Datenschutz in AWS Panorama. https://aws.amazon.com/compliance/shared-responsibility-model/ Wie in diesem Modell beschrieben, ist AWS für den Schutz der globalen Infrastruktur verantwortlich, in der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS-Modell der geteilten Verantwortung und in der DSGVO im AWS-Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit AWS Panorama oder anderen AWS-Services über die Konsole, APIAWS CLI, oder AWS SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Abschnitte

Verschlüsselung während der Übertragung

AWS Panorama-API-Endpunkte unterstützen sichere Verbindungen nur über HTTPS. Wenn Sie AWS Panorama-Ressourcen mit der AWS Management Console, dem AWS SDK oder der AWS Panorama-API verwalten, wird die gesamte Kommunikation mit Transport Layer Security (TLS) verschlüsselt. Die Kommunikation zwischen der AWS Panorama Appliance und AWS ist ebenfalls mit TLS verschlüsselt. Die Kommunikation zwischen der AWS Panorama Appliance und Kameras über RTSP ist nicht verschlüsselt.

Eine vollständige Liste der API-Endpunkte finden Sie unter AWS-Regionen und Endpunkte im Allgemeine AWS-Referenz.

AWS Panorama-Appliance

Die AWS Panorama-Appliance verfügt über physische Ports für Ethernet-, microSD-Video- und USB-Speicher. Der SD-Karten-Slot, das WLAN und Bluetooth sind nicht verwendbar. Der USB-Port wird nur während der Bereitstellung verwendet, um ein Konfigurationsarchiv an die Appliance zu übertragen.

Der Inhalt des Konfigurationsarchivs, das das Bereitstellungszertifikat und die Netzwerkkonfiguration der Appliance enthält, wird nicht verschlüsselt. AWS Panorama speichert diese Dateien nicht. Sie können nur abgerufen werden, wenn Sie eine Appliance registrieren. Nachdem Sie das Konfigurationsarchiv auf eine Appliance übertragen haben, löschen Sie es von Ihrem Computer und Ihrem USB-Speichergerät.

Das gesamte Dateisystem der Appliance ist verschlüsselt. Darüber hinaus wendet die Appliance mehrere Schutzmaßnahmen auf Systemebene an, darunter Rollback-Schutz für erforderliche Softwareupdates, signierten Kernel und Bootloader sowie Überprüfung der Softwareintegrität.

Wenn Sie die Appliance nicht mehr verwenden, führen Sie einen vollständigen Reset durch, um Ihre Anwendungsdaten zu löschen und die Appliance-Software zurückzusetzen.

Anwendungen

Sie steuern den Code, den Sie für Ihre Appliance bereitstellen. Validieren Sie den gesamten Anwendungscode vor der Bereitstellung auf Sicherheitsprobleme, unabhängig von der Quelle. Wenn Sie Bibliotheken von Drittanbietern in Ihrer Anwendung verwenden, sollten Sie die Lizenz- und Supportrichtlinien für diese Bibliotheken sorgfältig prüfen.

CPU-, Arbeitsspeicher- und Festplattennutzung von Anwendungen wird durch die Appliance-Software nicht eingeschränkt. Eine Anwendung, die zu viele Ressourcen verwendet, kann sich negativ auf andere Anwendungen und den Betrieb des Geräts auswirken. Testen Sie Anwendungen separat, bevor Sie Produktionsumgebungen kombinieren oder bereitstellen.

Anwendungsressourcen (Codes und Modelle) sind nicht vom Zugriff innerhalb Ihres Kontos, Ihrer Appliance oder Ihrer Build-Umgebung isoliert. Die von der AWS Panorama Application CLI generierten Container-Images und Modellarchive sind nicht verschlüsselt. Verwenden Sie separate Konten für Produktions-Workloads und erlauben Sie den Zugriff nur nach Bedarf.

Sonstige -Services

Um Ihre Modelle und Anwendungscontainer sicher in Amazon S3 zu speichern, verwendet AWS Panorama serverseitige Verschlüsselung mit einem Schlüssel, den Amazon S3 verwaltet. Weitere Informationen finden Sie unter Schutz von Daten durch Verschlüsselung im Benutzerhandbuch für Amazon Simple Storage Service.

Die Anmeldeinformationen für den Kamerastream werden im Ruhezustand in verschlüsseltAWS Secrets Manager. Die IAM-Rolle der Appliance erteilt ihr die Berechtigung, das Secret abzurufen, um auf den Benutzernamen und das Passwort des Streams zuzugreifen.

Die AWS Panorama Appliance sendet Protokolldaten an Amazon CloudWatch Logs. CloudWatch Logs verschlüsselt diese Daten standardmäßig und kann für die Verwendung eines vom Kunden verwalteten Schlüssels konfiguriert werden. Weitere Informationen finden Sie unter Verschlüsseln von Protokolldaten in - CloudWatch Protokollen mit AWS KMS im Amazon- CloudWatch Logs-Benutzerhandbuch.

Identity and Access Management für AWS Panorama

AWS Identity and Access Management (IAM) ist ein AWS-Service, mit dem Administratoren den Zugriff auf AWS-Ressourcen sicher steuern können. IAM-Administratoren steuern, wer für die Nutzung von AWS Panorama-Ressourcen authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) werden kann. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Themen

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, unterscheidet sich je nach Ihrer Arbeit in AWS Panorama.

Service-Benutzer – Wenn Sie den AWS Panorama-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie für Ihre Arbeit weitere AWS Panorama-Funktionen ausführen, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Featuresweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Wenn Sie nicht auf ein Feature in AWS Panorama zugreifen können, finden Sie weitere Informationen unter Fehlerbehebung für AWS Panorama-Identität und -Zugriff.

Service-Administrator – Wenn Sie in Ihrem Unternehmen für AWS Panorama-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollständigen Zugriff auf AWS Panorama. Ihre Aufgabe besteht darin, zu bestimmen, auf welche AWS Panorama-Funktionen und -Ressourcen Ihre Service-Benutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die

Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit AWS Panorama verwenden kann, finden Sie unter Funktionsweise von AWS Panorama mit IAM.

IAM-Administrator – Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf AWS Panorama verfassen können. Beispiele für identitätsbasierte AWS Panorama-Richtlinien, die Sie in IAM verwenden können, finden Sie unter Beispiele für identitätsbasierte AWS Panorama-Richtlinien.

Authentifizierung mit Identitäten

Authentifizierung ist die Art, wie Sie sich mit Ihren Anmeldeinformationen bei AWS anmelden. Die Authentifizierung (Anmeldung bei AWS) muss als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle erfolgen.

Sie können sich bei AWS als Verbundidentität mit Anmeldeinformationen anmelden, die über eine Identitätsquelle bereitgestellt werden. Benutzer von AWS IAM Identity Center (IAM Identity Center), die Single-Sign-on-Authentifizierung Ihres Unternehmens und Anmeldeinformationen für Google oder Facebook sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie auf AWS mithilfe des Verbunds zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der AWS Management Console oder beim AWS-Zugriffsportal anmelden. Weitere Informationen zum Anmelden bei AWS finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch von AWS-Anmeldung.

Bei programmgesteuertem Zugriff auf AWS bietet AWS ein Software Development Kit (SDK) und eine Command Line Interface (CLI, Befehlszeilenschnittstelle) zum kryptographischen Signieren Ihrer Anfragen mit Ihren Anmeldeinformationen. Wenn Sie keine AWS-Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode zum eigenen Signieren von Anforderungen finden Sie unter Signieren von AWS-API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise die Verwendung von Multi-Faktor Authentifizierung (MFA), um die Sicherheit Ihres Kontos zu verbessern. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center-Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto-Root-Benutzer

Wenn Sie ein AWS-Konto neu erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen des Kontos hat. Diese Identität wird als AWS-Konto-Root-Benutzer bezeichnet. Für den Zugriff auf den Root-Benutzer müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die zur Erstellung des Kontos verwendet wurden. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder eine einzelne Anwendung. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität in Ihrem AWS-Konto mit spezifischen Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können

vorübergehend eine IAM-Rolle in der AWS Management Console übernehmen, indem Sie Rollen wechseln. Sie können eine Rolle annehmen, indem Sie eine AWS CLI oder AWS-API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff: Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center-Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen: Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. In einigen AWS-Services können Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff: Einige AWS-Services verwenden Features in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward access sessions (FAS) – Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in AWS verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu

stellen. FAS-Anfragen werden nur dann gestellt, wenn ein Service eine Anfrage erhält, die eine Interaktion mit anderen AWS-Services oder -Ressourcen erfordert. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

- Servicerolle: Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
- Serviceverknüpfte Rolle: Eine serviceverknüpfte Rolle ist ein Typ von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene Rollen anzeigen, aber nicht bearbeiten.

- Anwendungen in Amazon EC2: Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und AWS CLI- oder AWS-API-Anforderungen durchführen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle einer EC2-Instance zuzuweisen und die Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Für die Zugriffssteuerung in AWS erstellen Sie Richtlinien und weisen diese den AWS-Identitäten oder -Ressourcen zu. Eine Richtlinie ist ein Objekt in AWS, das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und

Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Benutzerinformationen über die AWS Management Console, die AWS CLI oder die AWS -API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto anfügen können. Verwaltete Richtlinien umfassen von AWS verwaltete und von Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können

Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können verwaltete AWS-Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3, AWS WAF und Amazon VPC sind Beispiele für Dienste, die ACLs unterstützen. Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen:Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs) – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OE) in AWS Organizations angeben. AWS Organizations ist ein Dienst für die Gruppierung und zentrale Verwaltung mehrerer AWS-Konten Ihres Unternehmens. Wenn Sie innerhalb einer Organisation alle Features aktivieren,

können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. SCPs schränken Berechtigungen für Entitäten in Mitgliedskonten einschließlich des jeweiligen Root-Benutzer des AWS-Kontos ein. Weitere Informationen zu Organizations und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations-Benutzerhandbuch.

- Sitzungsrichtlinien:Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen dazu, wie AWS die Zulässigkeit einer Anforderung ermittelt, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter Logik für die Richtlinienauswertung im IAM-Benutzerhandbuch.

Funktionsweise von AWS Panorama mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf AWS Panorama zu verwalten, sollten Sie verstehen, welche IAM-Funktionen Sie mit AWS Panorama verwenden können. Einen Überblick über das Zusammenwirken von AWS Panorama und anderen -AWSServices mit IAM finden Sie unter -AWSServices, die mit IAM funktionieren im IAM-Benutzerhandbuch.

Eine Übersicht über die Berechtigungen, Richtlinien und Rollen, die von AWS Panorama verwendet werden, finden Sie unter AWS Panorama-Berechtigungen.

Beispiele für identitätsbasierte AWS Panorama-Richtlinien

Standardmäßig haben IAM-Benutzer und -Rollen keine Berechtigung zum Erstellen oder Ändern von AWS Panorama-Ressourcen. Sie können auch keine Aufgaben ausführen, die die AWS Management Console-, AWS CLI- oder AWS-API benutzen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den IAM-Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von Richtlinien auf der JSON-Registerkarte im IAM-Benutzerhandbuch.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der AWS Panorama-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand AWS Panorama-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen:Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete AWS-Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS-verwaltete Richtlinien oder AWS-verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten:Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs:Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn diese durch

ein bestimmtes AWS-Service, wie beispielsweise AWS CloudFormation, verwendet werden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten:IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.
- Bedarf einer Multi-Faktor-Authentifizierung (MFA):Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der AWS Panorama-Konsole

Um auf die AWS Panorama-Konsole zugreifen zu können, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den AWS Panorama-Ressourcen in Ihrem AWS Konto aufzulisten und anzuzeigen. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (IAM-Benutzer oder -Rollen) mit dieser Richtlinie.

Weitere Informationen finden Sie unter Identitätsbasierte IAM-Richtlinien für AWS Panorama.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

```
{
    "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": "*"
        }
    ]
}
```

AWSverwaltete Richtlinien für AWS Panorama

Eine von AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von AWS erstellt und verwaltet wird. Von AWS verwaltete Richtlinien stellen Berechtigungen für viele häufige Anwendungsfälle bereit, damit Sie beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS-verwaltete Richtlinien möglicherweise nicht die geringsten Berechtigungen für Ihre spezifischen Anwendungsfälle gewähren, da sie für alle AWS-Kunden verfügbar sind.

Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Die Berechtigungen, die in den von AWS verwalteten Richtlinien definiert sind, können nicht geändert werden. Wenn AWS Berechtigungen aktualisiert, die in einer von AWS verwalteten Richtlinie definiert werden, wirkt sich das Update auf alle Prinzipalidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWSaktualisiert am wahrscheinlichsten eine von AWS verwaltete Richtlinie, wenn ein neuer AWS-Service gestartet wird oder neue API-Operationen für bestehende Services verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS Panorama bietet die folgenden verwalteten Richtlinien. Den vollständigen Inhalt und den Änderungsverlauf der einzelnen Richtlinien finden Sie auf den verlinkten Seiten in der IAM-Konsole.

- AWSPanoramaFullAccess— Bietet vollen Zugriff auf AWS Panorama, AWS Panorama-Zugangspunkte in Amazon S3 und Appliance-Anmeldeinformationen inAWS Secrets Managerund Appliance-Logs bei AmazonCloudWatch. Beinhaltet die Erlaubnis zur Erstellung einesRolle im Zusammenhang mit Dienstleistungenfür AWS Panorama.
- AWSPanoramaServiceLinkedRolePolicy— Ermöglicht AWS Panorama die Verwaltung von Ressourcen in AWS IoT, AWS Secrets Manager und AWS Panorama.
- AWSPanoramaApplianceServiceRolePolicy— Ermöglicht einer AWS Panorama-Appliance das Hochladen von Protokollen aufCloudWatchund um Objekte von Amazon S3-Access Points abzurufen, die von AWS Panorama erstellt wurden.

AWS Panorama wird aktualisiert aufAWSverwaltete Richtlinien

In der folgenden Tabelle werden Aktualisierungen der verwalteten Richtlinien für AWS Panorama beschrieben.

Änderung	Beschreibung	Datum
AWSPanoramaFullAccess – Aktualisierung auf eine bestehende Richtlinie	Der Benutzerrichtlinie wurden Berechtigungen hinzugefügt, um Benutzern das Anzeigen von Protokollgruppen in	20.01.13

Änderung	Beschreibung	Datum
	derCloudWatchProtokollierkonsole.	
AWSPanoramaFullAccess – Aktualisierung auf eine bestehende Richtlinie	Der Benutzerrichtlinie wurden Berechtigungen hinzugefügt, damit Benutzer das AWS-Panorama verwalten könnenRolle im Zusammenhang mit Dienstleistungenund um auf AWS Panorama-Ressourcen in anderen Diensten wie IAM, Amazon S3 zuzugreifen,CloudWatchund Secrets Manager.	20.10.2021
AWSPanoramaApplianceServiceRolePolicy – Neue Richtlinie.	Neue Richtlinie für die Servicerolle AWS Panorama Appliance	20.10.2021
AWSPanoramaServiceLinkedRolePolicy – Neue Richtlinie.	Neue Richtlinie für die serviceverknüpfte Rolle von AWS Panorama.	20.10.2021
AWS Panorama hat begonnen, Änderungen zu verfolgen	AWS Panorama begann mit der Nachverfolgung von Änderungen fürAWSverwaltete Richtlinien.	20.10.2021

Verwenden von serviceverknüpften Rollen für AWS Panorama

AWS Panorama verwendet serviceverknüpfte Rollen von AWS Identity and Access Management (IAM). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit AWS Panorama verknüpft ist. Serviceverknüpfte Rollen werden von AWS Panorama vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von AWS Panorama, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. AWS Panorama definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur AWS Panorama die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre AWS Panorama-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS-Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Yes (Ja) aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Abschnitte

Berechtigungen von serviceverknüpften Rollen für AWS Panorama

AWS Panoramaverwendet die serviceverknüpfte Rolle namens .awsServiceRoleforawSpanorama—Ermöglicht AWS Panorama, Ressourcen in AWS IoT, AWS Secrets Manager und AWS Panorama zu verwalten.

Die serviceverknüpfte Rolle AWSServiceRoleForAWSSpanorama vertraut, dass die folgenden Services die Rolle übernehmen:

- `panorama.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt AWS Panorama die Durchführung der folgenden Aktionen:

- MonitorAWS PanoramaRessourcen

- VerwaltenAWS IoTressourcen für dasAWS PanoramaAppliance
- Zugriff aufAWS Secrets ManagerGeheimnisse um Kamera-Anmeldeinformationen zu erhalten

Eine vollständige Liste der Berechtigungen finden Siedie awspanoramaserviceLinkedRolePolicy Richtlinie anzeigenin der IAM-Konsole.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für AWS Panorama

Sie müssen eine servicegebundene Rolle nicht manuell erstellen. Wenn Sie eine Appliance imAWS Management Console, derAWS CLIoder dasAWSAPI,AWS Panoramaerstellt die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine Appliance registrieren,AWS Panoramaerstellt erneut die serviceverknüpfte Rolle für Sie.

Bearbeiten einer serviceverknüpften Rolle für AWS Panorama

AWS Panoramaberechtigt Sie nicht zum Bearbeiten der serviceverknüpften Rolle AWSServiceRoleForAWSSpannorama. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer servicegebundenen Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für AWS Panorama

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

So löschen Sie dasAWS Panorama-Ressourcen, die von AWSServiceRoleForAWSStorama verwendet werden, verwenden Sie die Verfahren in den folgenden Abschnitten dieses Leitfadens.

- Versionen und Anwendungen löschen

- Einen Appliance abmelden

> **Note**
>
> Wenn der AWS Panorama-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Verwenden Sie die IAM-Konsole, die serviceverknüpfte Rolle AWSServiceRoleForAWSSpannoramaAWS CLIoder dasAWSAPI. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für serviceverknüpfte AWS Panorama-Rollen

AWS Panorama unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unterAWSRegionen und Endpunkteaus.

Dienstübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine juristische Stelle, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiertere juristische Stelle zwingen kann, die Aktion auszuführen. In AWS kann der dienstübergreifende Identitätswechsel zu Confused-Deputy-Problem führen. Ein serviceübergreifender Identitätswechsel kann auftreten, wenn ein Service (der aufrufende Service) einen anderen Service aufruft (der aufgerufene Service). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen die Verwendung der globalen Bedingungskontext-Schlüsseln `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen, die AWS Panorama einem anderen Service erteilt, auf eine bestimmte Ressource zu beschränken. Wenn Sie beide globalen Bedingungskontextschlüssel verwenden, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden.

Der Wert von`aws:SourceArn`muss der ARN einesAWS Panorama-Gerät.

Der effektivste Weg, um sich vor dem verwirrten Stellvertreterproblem zu schützen, ist die Verwendung des`aws:SourceArn`globaler Kontextschlüssel mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht wissen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den`aws:SourceArn`globaler Kontextbedingungsschlüssel mit Platzhaltern (`*`) für die unbekannten Teile des ARN. Zum Beispiel, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

Anweisungen zum Sichern der Servicerolle, dieAWS Panoramaverwendet, um die Erlaubnis zu erteilenAWS PanoramaAppliance, sieheAbsichern der Appliance-Rolleaus.

Fehlerbehebung für AWS Panorama-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS Panorama und IAM auftreten können.

Themen

- Ich bin nicht autorisiert, eine Aktion in AWS Panorama auszuführen
- Ich bin nicht autorisiert, iam durchzuführen:PassRole
- Ich möchte Personen außerhalb meines -AWSKontos Zugriff auf meine AWS Panorama-Ressourcen gewähren

Ich bin nicht autorisiert, eine Aktion in AWS Panorama auszuführen

Wenn die AWS Management Console Ihnen mitteilt, dass Sie nicht zur Ausführung einer Aktion autorisiert sind, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM-Benutzer versucht, die Konsole zu verwenden, um Details zu einer Appliance anzuzeigen, aber keine `panorama:DescribeAppliance` Berechtigungen hat.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 panorama:DescribeAppliance on resource: my-appliance
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion `my-appliance` auf die Ressource `panorama:DescribeAppliance` zugreifen zu können.

Ich bin nicht autorisiert, iam durchzuführen:PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Ausführung der `iam:PassRole` Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an AWS Panorama übergeben zu können.

Einige AWS-Services erlauben die Übergabe einer vorhandenen Rolle an diesen Dienst, sodass keine neue Servicerolle oder serviceverknüpfte Rolle erstellt werden muss. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS Panorama auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines -AWSKontos Zugriff auf meine AWS Panorama-Ressourcen gewähren

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob AWS Panorama diese Funktionen unterstützt, finden Sie unter Funktionsweise von AWS Panorama mit IAM.
- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konto im IAM-Benutzerhandbuch.

- Informationen dazu, wie Sie AWS-Konten-Drittanbieter Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter Gewähren des Zugriffs auf AWS-Konten von externen Benutzern im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

Konformitätsvalidierung für AWS Panorama

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen zu Sicherheit und Compliance — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- Architecting for HIPAA Security and Compliance on Amazon Web Services — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-fähige Anwendungen erstellen AWS können.

  > **Note**
  >
  > AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der Referenz für HIPAA-berechtigte Services.

- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.

- AWS Security Hub— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerungsreferenz.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Zusätzliche Überlegungen in Bezug auf die Anwesenheit von Personen

Im Folgenden finden Sie einige bewährte Methoden, die Sie bei der Verwendung von AWS Panorama für Szenarien berücksichtigen sollten, in denen Personen anwesend sein könnten:

- Stellen Sie sicher, dass Sie alle geltenden Gesetze und Vorschriften für Ihren Anwendungsfall kennen und einhalten. Dazu gehören möglicherweise Gesetze in Bezug auf die Positionierung und das Sichtfeld Ihrer Kameras, Anforderungen an Hinweise und Beschilderungen bei der Platzierung und Verwendung von Kameras sowie die Rechte von Personen, die in Ihren Videos möglicherweise anwesend sind, einschließlich ihrer Datenschutzrechte.
- Berücksichtigen Sie die Auswirkungen Ihrer Kameras auf Menschen und deren Privatsphäre. Überlegen Sie sich zusätzlich zu den gesetzlichen Anforderungen, ob es angemessen wäre, in Bereichen, in denen sich Ihre Kameras befinden, einen Hinweis anzubringen, und ob Kameras gut sichtbar und frei von Verdeckungen angebracht werden sollten, damit die Leute nicht überrascht sind, dass sie möglicherweise vor der Kamera stehen.
- Halten Sie geeignete Richtlinien und Verfahren für den Betrieb Ihrer Kameras und die Überprüfung der von den Kameras gewonnenen Daten bereit.
- Erwägen Sie angemessene Zugriffskontrollen, Nutzungsbeschränkungen und Aufbewahrungsfristen für die von Ihren Kameras erfassten Daten.

Infrastruktursicherheit in AWS Panorama

Als verwalteter Service ist AWS Panorama geschützt durchAWSglobale Netzwerksicherheit. Informationen zu AWS-Sicherheitsdiensten und wie AWS die Infrastruktur schützt, finden Sie unter AWSCloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastrukturschutz im Security Pillar AWS Well-Architected Framework.

Du verwendestAWSveröffentlichte API-Aufrufe für den Zugriff auf AWS Panorama über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Bereitstellung der AWS Panorama Appliance in Ihrem Rechenzentrum

Die AWS-Panorama-Appliance benötigt Internetzugang, um mitAWSDienste. Es benötigt auch Zugriff auf Ihr internes Kameranetzwerk. Es ist wichtig, dass Sie Ihre Netzwerkkonfiguration sorgfältig abwägen und jedem Gerät nur den Zugriff gewähren, den es benötigt. Seien Sie vorsichtig, wenn Ihre Konfiguration es der AWS Panorama Appliance ermöglicht, als Brücke zu einem sensiblen IP-Kameranetzwerk zu fungieren.

Sie sind für Folgendes verantwortlich:

- Die physische und logische Netzwerksicherheit der AWS Panorama Appliance.
- Sicherer Betrieb der an das Netzwerk angeschlossenen Kameras, wenn Sie die AWS Panorama Appliance verwenden.
- Halten Sie die AWS-Panorama-Appliance und die Kamerasoftware auf dem neuesten Stand.
- Einhaltung aller geltenden Gesetze oder Vorschriften in Bezug auf den Inhalt der Videos und Bilder, die Sie in Ihren Produktionsumgebungen sammeln, einschließlich solcher, die sich auf den Datenschutz beziehen.

Die AWS Panorama Appliance verwendet unverschlüsselte RTSP-Kamerastreams. Weitere Informationen zum Verbinden der AWS Panorama Appliance mit Ihrem Netzwerk finden Sie unterDie AWS Panorama Appliance mit Ihrem Netzwerk verbinden. Einzelheiten zur Verschlüsselung finden Sie unterDatenschutz in AWS Panorama.

Laufzeitumgebungssoftware in AWS Panorama

AWS Panorama bietet Software, die Ihren Anwendungscode in einer Ubuntu Linux basierten Umgebung auf der AWS Panorama Appliance ausführt. AWS Panorama ist dafür verantwortlich, die Software im Appliance-Image auf dem neuesten Stand zu halten. AWS Panorama veröffentlicht regelmäßig Softwareupdates, die Sie anwenden könnenVerwenden der AWS Panorama Panorama-Konsoleaus.

Sie können Bibliotheken in Ihrem Anwendungscode verwenden, indem Sie sie in der Anwendung installieren`Dockerfile`aus. Um die Anwendungsstabilität zwischen Builds sicherzustellen, wählen Sie eine bestimmte Version jeder Bibliothek aus. Aktualisieren Sie Ihre Abhängigkeiten regelmäßig, um Sicherheitsprobleme zu beheben.

Versionen

Die folgende Tabelle zeigt, wann Funktionen und Softwareupdates für den AWS Panorama Service, die Software und die Dokumentation veröffentlicht wurden. Um sicherzustellen, dass Sie Zugriff auf alle Funktionen haben, aktualisieren Sie Ihre AWS Panorama Appliance auf die neueste Softwareversion. Weitere Informationen zu einer Version finden Sie im verknüpften Thema.

Änderung	Beschreibung	Datum
Aktualisierung der Appliance-Software	Version 7.0.13 ist ein Hauptversions-Update, das die Verwaltung von Software-Updates durch die Appliance ändert. Wenn Sie die ausgehende Netzwerkkommunikation von der Appliance einschränken oder sie mit einem privaten VPC-Subnetz verbinden, müssen Sie den Zugriff auf zusätzliche Endpunkte und Ports zulassen, bevor Sie das Update anwenden. Weitere Informationen finden Sie im Änderungsprotokoll .	28. Dezember 2023
Aktualisierung der Appliance-Software	Version 6.2.1 enthält Fehlerbehebungen. Weitere Informationen finden Sie im Änderungsprotokoll .	6. September 2023
Aktualisierung der Appliance-Software	Version 6.0.8 enthält Fehlerbehebungen und Sicherheitsverbesserungen. Weitere Informationen finden Sie im Änderungsprotokoll .	6. Juli 2023

Aktualisierung der Appliance-Software	Version 5.1.7 enthält Fehlerbehebungen und Verbesserungen bei der Fehlerbehandlung. Weitere Informationen finden Sie im Änderungsprotokoll .	31. März 2023
Konsolenaktualisierung	Sie können die AWS Panorama -Appliance jetzt über die -Managementkonsole kaufen. Informationen zum Erteilen der Berechtigung für einen Benutzer zum Kauf von Geräten finden Sie unter Identitätsbasierte IAM-Richtlinien für AWS Panorama.	2. Februar 2023
Aktualisierung der Appliance-Software	Version 5.0.74 enthält Fehlerbehebungen und Verbesserungen bei der Fehlerbehandlung. Weitere Informationen finden Sie im Änderungsprotokoll .	23. Januar 2023
API-Update	`AllowMajorVersionUpdate` Option hinzugefügt `OTAJobConfig` , um die Anmeldung für Hauptversionsaktualisierungen der Appliance-Software zu ermöglichen. Weitere Informationen finden Sie unter CreateJobForDevices.	19. Januar 2023

Neues Tool für Entwickler	Das neue Tool „Sideloading“ ist im AWS Panorama Beispiel- GitHub Repository verfügbar. Sie können dieses Tool verwenden, um Anwendungscode zu aktualisieren, ohne einen Container zu erstellen und bereitzustellen. Weitere Informationen finden Sie unter README.	16. November 2022
Aktualisierung des Anwendungs-Basis-Images	Version 1.2.0 fügt eine Timeout-Option zu hinzu`video_in.get()` , legt die `AWS_REGION` Umgebungsvariable fest und verbessert die Fehlerbehandlung. Weitere Informationen finden Sie im Änderungsprotokoll .	16. November 2022
Aktualisierung der Appliance-Software	Version 5.0.42 enthält Fehlerbehebungen und Sicherheitsupdates. Weitere Informationen finden Sie im Änderungsprotokoll .	16. November 2022
Aktualisierung der Appliance-Software	Version 5.0.7 bietet Unterstützung für das Remote-Neustarten von Appliances und das Remote-Pausieren von Kamerastreams. Weitere Informationen finden Sie im Änderungsprotokoll .	13. Oktober 2022

Aktualisierung der Appliance-Software	Version 4.3.93 bietet Unterstützung für das Abrufen von Protokollen von einem Offline-Gerät . Weitere Informationen finden Sie im Änderungsprotokoll .	24. August 2022
Aktualisierung der Appliance-Software	Version 4.3.72 enthält Fehlerbehebungen und Sicherheitsupdates. Weitere Informationen finden Sie im Änderungsprotokoll .	23. Juni 2022
AWS PrivateLink--Support	AWS Panorama unterstützt VPC-Endpunkte für die Verwaltung von AWS Panorama Ressourcen aus einem privaten Subnetz. Weitere Informationen finden Sie unter Verwenden von VPC-Endpunkten.	2. Juni 2022
Aktualisierung der Appliance-Software	Version 4.3.55 verbessert die Speicherauslastung für das `console_output` Protokoll . Weitere Informationen finden Sie im Änderungsprotokoll .	5. Mai 2022

Bol SE ThinkEdgeSE70	Eine neue Appliance für AWS Panorama ist über Bol verfügbar. Das Bol ThinkEdge SE70, das von Nvidia Jetson Xavier NX unterstützt wird, unterstützt dieselben Funktionen wie das AWS Panorama -Gerät. Weitere Informationen finden Sie unter Kompatible Geräte.	6. April 2022
Aktualisierung des Anwendungs-Basis-Images	Version 1.1.0 verbessert die Leistung beim Ausführen von Hintergrund-Threads und fügt Medienobjekten ein Flag (is_cached) hinzu, das angibt, ob das Image aktuell ist. Weitere Informationen finden Sie unter gallery.ecr.aws.	29. März 2022
Aktualisierung der Appliance-Software	Version 4.3.45 bietet Unterstützung für GPU-Zugriff und eingehende Ports . Weitere Informationen finden Sie im Änderungsprotokoll .	24. März 2022
Aktualisierung der Appliance-Software	Version 4.3.35 verbessert die Sicherheit und Leistung. Weitere Informationen finden Sie im Änderungsprotokoll .	22. Februar 2022

Aktualisierte verwaltete Richtlinien	AWS Identity and Access Management Von verwaltete Richtlinien für AWS Panorama wurden aktualisiert. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien.	13. Januar 2022
Bereitstellen von Protokollen	Mit Appliance-Software 4.3.23 schreibt die Appliance während der Bereitstellung Protokolle auf ein USB-Laufwerk. Weitere Informationen finden Sie unter Protokolle.	13. Januar 2022
NTP-Serverkonfiguration	Sie können die AWS Panorama Appliance jetzt so konfigurieren, dass sie einen bestimmten NTP-Server für die Taktsynchronisierung verwendet. Konfigurieren Sie NTP-Einstellungen während der Appliance-Einrichtung mit anderen Netzwerkeinstellungen. Weitere Informationen finden Sie unter Einrichten von .	13. Januar 2022
Zusätzliche Regionen	AWS Panorama ist jetzt in den Regionen Asien-Pazifik (Singapur) und Asien-Pazifik (Sydney) verfügbar.	13. Januar 2022

Aktualisierung der Appliance-Software	Version 4.3.4 bietet Unterstützung für die `-precision Mode` Einstellung für Modelle und aktualisiert das Protokollierungsverhalten. Weitere Informationen finden Sie im Änderungsprotokoll .	8. November 2021
Von verwaltete Richtlinien aktualisiert	AWS Identity and Access Management Von verwaltete Richtlinien für AWS Panorama wurden aktualisiert. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien.	20. Oktober 2021
Allgemeine Verfügbarkeit	AWS Panorama ist jetzt für alle Kunden in den Regionen USA Ost (Nord-Virginia), USA West (Oregon), Europa (Irland) und Kanada (Zentral) verfügbar. Um eine AWS Panorama -Appliance zu kaufen, besuchen Sie AWS Panorama.	20. Oktober 2021
Vorversion	AWS Panorama ist auf Einladung in den Regionen USA Ost (Nord-Virginia) und USA West (Oregon) verfügbar.	1. Dezember 2020

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.