Guida per gli sviluppatori

AWS Panorama

AWS Panorama: Guida per gli sviluppatori

Table of Contents

Cos'è AWS Panorama?

AWS Panoramaè un servizio che porta la visione artificiale nella rete di telecamere locale. Installing del pluginAWS PanoramaAppliance o un altro dispositivo compatibile nel tuo datacenter, registralo conAWS Panoramae distribuisci applicazioni di visione artificiale dal cloud.AWS Panoramafunziona con le telecamere di rete RTSP (Real Time Streaming Protocol) esistenti. L'apparecchio esegue applicazioni di visione artificiale sicure daAWSPartnero applicazioni create da voi con ilAWS PanoramaSDK dell'applicazione.

LaAWS PanoramaAppliance è un dispositivo edge compatto che utilizza un potente system-on-module (SOM) ottimizzato per i carichi di lavoro di machine learning. L'apparecchio può eseguire più modelli di visione artificiale su più flussi video in parallel e produrre i risultati in tempo reale. È progettato per l'uso in ambienti commerciali e industriali ed è classificato per la protezione da polvere e liquidi (IP-62).

LaAWS PanoramaAppliance consente di eseguire applicazioni di visione artificiale autonome all'edge, senza inviare immagini al cloud AWS. Utilizzando l'SDK AWS, puoi integrarti con altri servizi AWS e utilizzarli per tenere traccia dei dati dell'applicazione nel tempo. Integring con altri servizi AWS, è possibile utilizzareAWS Panoramaper eseguire queste operazioni:

- Analisi dei modelli di trafficoutilizzo dell'SDK AWS per registrare dati per l'analisi di vendita al dettaglio in Amazon DynamoDB. Usa un'applicazione serverless per analizzare i dati raccolti nel tempo, rilevare anomalie nei dati e prevedere il comportamento future.
- Ricevi avvisi sulla sicurezza del sito— Monitorare le aree off-limits in un sito industriale. Se l'applicazione rileva una situazione potenzialmente pericolosa, carica un'immagine in Amazon Simple Storage Service (Amazon S3) ed eventualmente invia una notifica a un argomento Amazon Simple Notification Service (Amazon SNS) in modo che i destinatari possano eseguire le operazioni correttive.
- Migliora il controllo della qualità— Monitorare l'output di una linea di assemblaggio per identificare le parti che non sono conformi ai requisiti. Evidenzia le immagini di parti non conformi con testo e un riquadro di delimitazione e visualizzale su un monitor per la revisione da parte del tuo team di controllo qualità.
- Raccogli dati di formazione e test- Carica immagini di oggetti che il tuo modello di visione artificiale non è stato in grado di identificare o in cui la fiducia del modello nella sua ipotesi era al limite. Usa un'applicazione serverless per creare una coda di immagini che devono essere taggate. Contrassegna le immagini e usale per riqualificare il modello in Amazon SageMaker.

AWS Panoramautilizza altri servizi AWS per gestireAWS PanoramaAppliance, accesso a modelli e codice e distribuzione di applicazioni.AWS Panoramafa il più possibile senza richiedere all'utente di interagire con altri servizi, ma una conoscenza dei seguenti servizi può aiutarti a capire comeAWS Panoramafunziona.

- SageMaker— È possibile utilizzare SageMaker raccogliere dati di training da videocamere o sensori, creare un modello di machine learning e addestrarlo per la visione artificiale.AWS Panoramautilizza SageMaker Neo per ottimizzare i modelli da eseguire sulAWS PanoramaAppliance.
- Amazon S3— I punti di accesso di Amazon S3 vengono utilizzati per lo stage del codice dell'applicazione, dei modelli e dei file di configurazione per la distribuzione in unAWS PanoramaAppliance.
- AWS IoT–AWS PanoramautilizzaAWS IoTservizi per monitorare lo stato delAWS PanoramaAppliance, gestione degli aggiornamenti software e distribuzione di applicazioni. Non è necessario utilizzareAWS IoTdirettamente.

Per iniziare a utilizzare il pluginAWS PanoramaAppliance e scopri ulteriori informazioni sul servizio, continua a utilizzareNozioni di base su AWS Panorama.

Nozioni di base su AWS Panorama

Per iniziareAWS Panorama, scopri innanzitutto i concetti del servizio e la terminologia utilizzata in questa guida. Quindi puoi utilizzare laAWS Panorama console per registrare il tuoAWS Panorama dispositivo e creare un'applicazione. In circa un'ora, puoi configurare il dispositivo, aggiornarne il software e distribuire un'applicazione di esempio. Per completare i tutorial in questa sezione, si utilizzano l'AWS Panoramaappliance e una videocamera che trasmette video in streaming su una rete locale.

> **Note**
>
> Per acquistare unAWS Panorama dispositivo, accedi allaAWS Panorama console.

L'applicazioneAWS Panorama di esempio dimostra l'uso delleAWS Panorama funzionalità. Include un modello che è stato addestrato SageMaker e un codice di esempio che utilizza l'AWS PanoramaApplication SDK per eseguire inferenze e generare video. L'applicazione di esempio include unAWS CloudFormation modello e degli script che mostrano come automatizzare i flussi di lavoro di sviluppo e distribuzione dalla riga di comando.

Gli ultimi due argomenti di questo capitolo descrivono in dettaglio i requisiti per i modelli e le fotocamere e le specifiche hardwareAWS Panorama dell'accessorio. Se non avete ancora acquistato un apparecchio e delle fotocamere o avete intenzione di sviluppare modelli di visione artificiale personalizzati, consultate prima questi argomenti per ulteriori informazioni.

Argomenti

I concetti di AWS Panorama

In AWS Panorama, crei applicazioni di visione artificiale e le distribuisci su AWS Panorama Appliance o su un dispositivo compatibile per analizzare i flussi video dalle telecamere di rete. Scrivi codice applicativo in Python e crei contenitori di applicazioni con Docker. Utilizzi l'interfaccia a riga di comando di AWS Panorama Application per importare modelli di machine learning localmente o da Amazon Simple Storage Service (Amazon S3). Le applicazioni utilizzano l'SDK dell'applicazione AWS Panorama per ricevere input video da una telecamera e interagire con un modello.

Concetti

- L'appliance AWS Panorama
- Dispositivi compatibili
- Applicazioni
- Nodi
- Modelli

L'appliance AWS Panorama

L'AWS Panorama Appliance è l'hardware che esegue le tue applicazioni. La console AWS Panorama viene utilizzata per registrare un'appliance, aggiornarne il software e distribuirvi applicazioni. Il software su AWS Panorama Appliance si collega ai flussi delle telecamere, invia frame di video all'applicazione e visualizza l'output video su un display collegato.

AWS Panorama Appliance è undispositivo perimetrale alimentato da Nvidia Jetson AGX Xavier. Invece di inviare immagini aAWSCloud per l'elaborazione, esegue le applicazioni localmente su hardware ottimizzato. Ciò consente di analizzare i video in tempo reale ed elaborare i risultati localmente. L'appliance richiede una connessione Internet per segnalare lo stato, caricare i registri ed eseguire aggiornamenti e distribuzioni software.

Per ulteriori informazioni, consulta Gestione diAWS PanoramaAppliance.

Dispositivi compatibili

Oltre ad AWS Panorama Appliance, AWS Panorama supporta dispositivi compatibili daAWSPartner. I dispositivi compatibili supportano le stesse funzionalità di AWS Panorama Appliance. Registri e gestisci dispositivi compatibili con la console e l'API AWS Panorama e crei e distribuisci applicazioni nello stesso modo.

- Lenovo ThinkEdge® SE 70— Basato su Nvidia Jetson Xavier NX

I contenuti e le applicazioni di esempio di questa guida sono sviluppati con AWS Panorama Appliance. Per ulteriori informazioni su caratteristiche hardware e software specifiche per il tuo dispositivo, consulta la documentazione del produttore.

Applicazioni

Le applicazioni vengono eseguite su AWS Panorama Appliance per eseguire attività di visione artificiale su flussi video. Puoi creare applicazioni di visione artificiale combinando codice Python e modelli di apprendimento automatico e distribuirle su AWS Panorama Appliance tramite Internet. Le applicazioni possono inviare video a un display o utilizzare l'SDK AWS per inviare risultati ai servizi AWS.

Per creare e distribuire applicazioni, utilizzi l'interfaccia a riga di comando dell'applicazione AWS Panorama. L'interfaccia a riga di comando di AWS Panorama Application è uno strumento a riga di comando che genera cartelle applicative e file di configurazione predefiniti, crea contenitori con Docker e carica risorse. Puoi eseguire più applicazioni su un unico dispositivo.

Per ulteriori informazioni, consulta Gestione diAWS Panoramaapplicazioni.

Nodi

Un'applicazione è composta da più componenti denominatinodi, che rappresentano input, output, modelli e codice. Un nodo può essere solo configurato (ingressi e uscite) o includere artefatti (modelli e codice). Il nodo di codice di un'applicazione è inclusopacchetti di nodiche carichi su un punto di accesso Amazon S3, a cui può accedervi l'AWS Panorama Appliance. Unmanifesto dell'applicazioneè un file di configurazione che definisce le connessioni tra i nodi.

Per ulteriori informazioni, consulta Nodi di applicazione.

Modelli

Un modello di visione artificiale è una rete di apprendimento automatico addestrata per elaborare immagini. I modelli di visione artificiale possono eseguire varie attività come la classificazione, il rilevamento, la segmentazione e il tracciamento. Un modello di visione artificiale acquisisce un'immagine come input e fornisce informazioni sull'immagine o sugli oggetti in essa contenuti.

AWS Panorama supporta modelli creati con PyTorch, Apache MXNet e TensorFlow. Puoi creare modelli con Amazon SageMaker o nel tuo ambiente di sviluppo. Per ulteriori informazioni, consulta ???.

Configurare AWS Panorama Appliance

Per iniziare a utilizzare la tua appliance AWS Panorama o un dispositivo compatibile, registralo nella console AWS Panorama e aggiorna il software. Durante il processo di configurazione, crei una risorsa dell'appliance in AWS Panorama che rappresenta l'appliance fisica e copi i file sull'appliance con un'unità USB. L'appliance utilizza questi certificati e file di configurazione per connettersi al servizio AWS Panorama. Quindi usi la console AWS Panorama per aggiornare il software dell'appliance e registrare le telecamere.

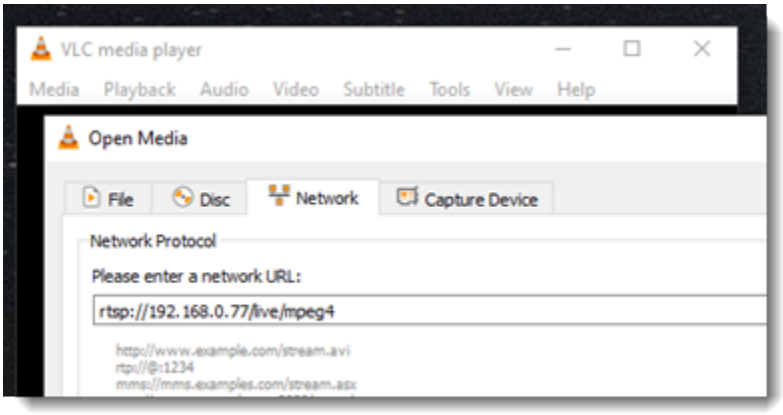
Sezioni

Prerequisiti

Per seguire AWS Panorama Appliance, è necessario disporre di AWS Panorama.

- Display: un display con ingresso HDMI per visualizzare l'output dell'applicazione di esempio.
- Unità USB (inclusa in AWS Panorama Appliance): un'unità di memoria flash USB 3.0 in formato FAT32 con almeno 1 GB di storage, per trasferire un archivio con file di configurazione e un certificato all'AWS Panorama Appliance.
- Videocamera: una telecamera IP che emette un flusso video RTSP.

Utilizza gli strumenti e le istruzioni forniti dal produttore della videocamera per identificare l'indirizzo IP e il percorso di streaming della telecamera. Puoi utilizzare un lettore video come VLC per verificare l'URL dello streaming, aprendolo come fonte multimediale di rete:



La console AWS Panorama utilizza altri servizi AWS per assemblare componenti applicativi, gestire le autorizzazioni e verificare le impostazioni. Per registrare un dispositivo e distribuire l'applicazione di esempio, sono necessarie le seguenti autorizzazioni:

- AWSPanoramaFullAccess— Fornisce accesso completo ad AWS Panorama, ai punti di accesso AWS Panorama in Amazon S3, alle credenziali dell'appliance e ai registri delle appliance in AWS Secrets Manager Amazon. CloudWatch Include l'autorizzazione a creare un ruolo collegato ai servizi per AWS Panorama.
- AWS Identity and Access Management(IAM): alla prima esecuzione, per creare ruoli utilizzati dal servizio AWS Panorama e dall'appliance AWS Panorama.

Se non disponi dell'autorizzazione per creare ruoli in IAM, chiedi a un amministratore di aprire la console AWS Panorama e di accettare la richiesta di creare ruoli di servizio.

Registra e configura l'appliance AWS Panorama

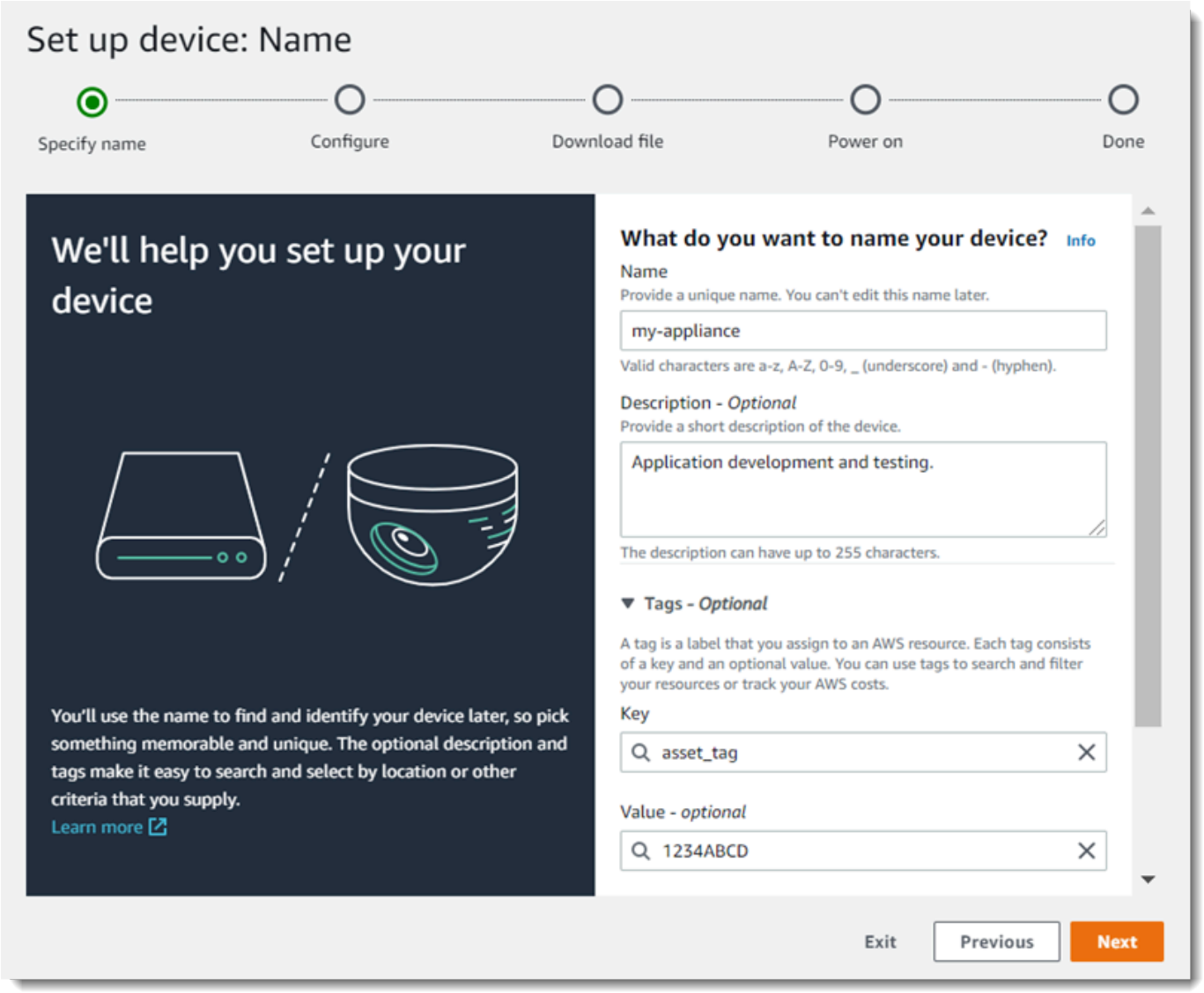
L'appliance AWS Panorama è un dispositivo hardware che si collega a telecamere abilitate alla rete tramite una connessione di rete locale. Utilizza un sistema operativo basato su Linux che include l'SDK per applicazioni AWS Panorama e il software di supporto per l'esecuzione di applicazioni di visione artificiale.

Per connettersi AWS per la gestione e l'installazione dell'applicazione, l'appliance utilizza un certificato del dispositivo. Utilizzi la console AWS Panorama per generare un certificato di provisioning. L'appliance utilizza questo certificato temporaneo per completare la configurazione iniziale e scaricare un certificato permanente del dispositivo.

> **Important**
>
> Il certificato di provisioning generato in questa procedura è valido solo per 5 minuti. Se non completi il processo di registrazione entro questo periodo di tempo, devi ricominciare da capo.

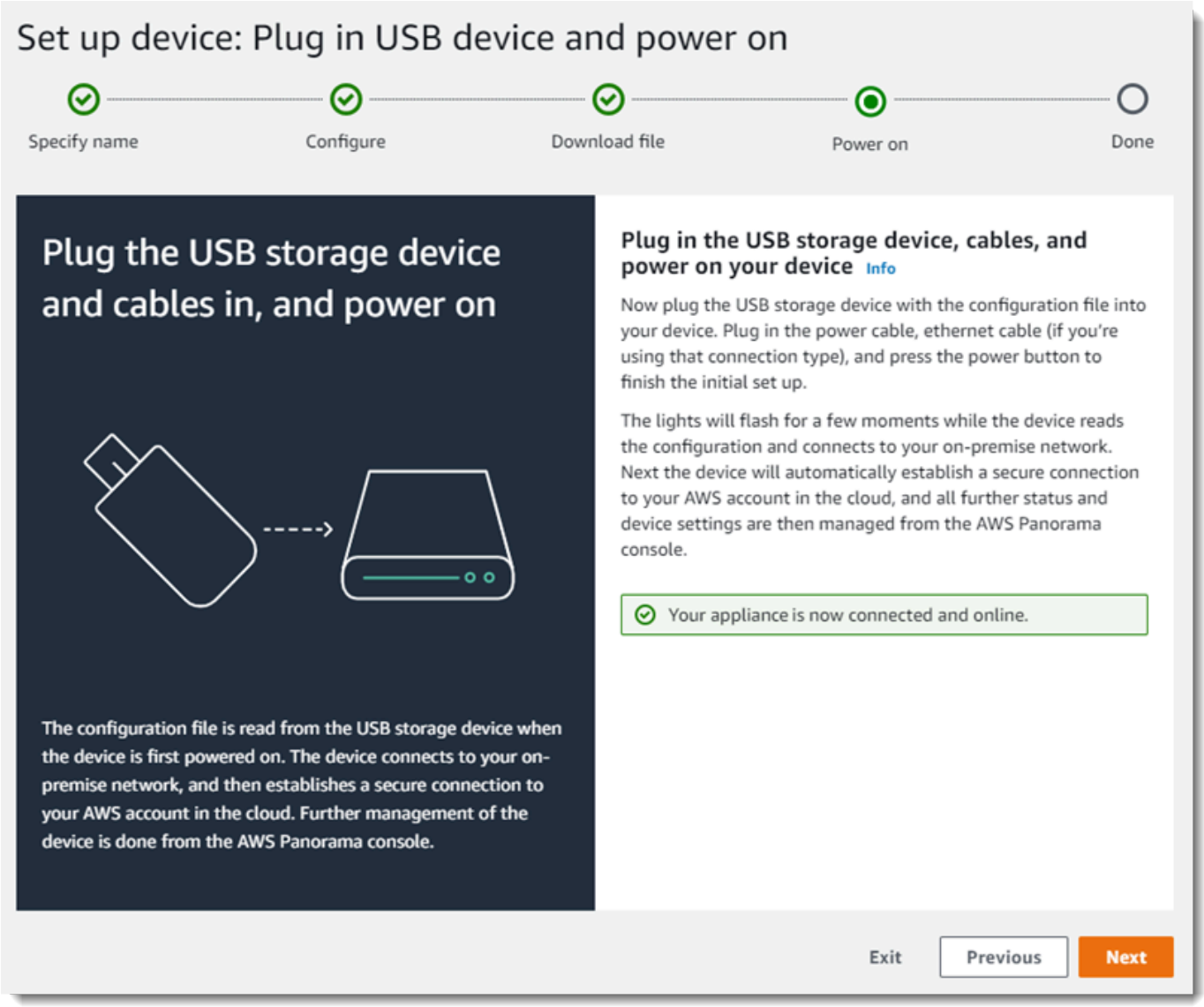
Per registrare un elettrodomestico

1. Connect l'unità USB al computer. Preparare l'apparecchio collegando la rete e i cavi di alimentazione. L'appliance si accende e attende il collegamento di un'unità USB.
2. Apri la pagina introduttiva della console AWS Panorama.
3. Scegli Aggiungi dispositivo.
4. Scegli Inizia configurazione.
5. Immettere AWS Panorama. Seleziona Next (Successivo).



6. Se devi assegnare manualmente un indirizzo IP, un server NTP o le impostazioni DNS, scegli Impostazioni di rete avanzate. Altrimenti, scegli Next (Successivo).
7. Scegli Scarica archivio. Seleziona Successivo.
8. Copia l'archivio di configurazione nella directory principale dell'unità USB.
9. Connect l'unità USB alla porta USB 3.0 sulla parte anteriore dell'accessorio, accanto alla porta HDMI.

   Quando si collega l'unità USB, l'appliance copia l'archivio di configurazione e il file di configurazione della rete su se stesso e si connette al AWS cloud. La spia di stato dell'apparecchiatura passa da verde a blu mentre completa la connessione, quindi torna verde.
10. Scegliere Next (Avanti) per continuare.



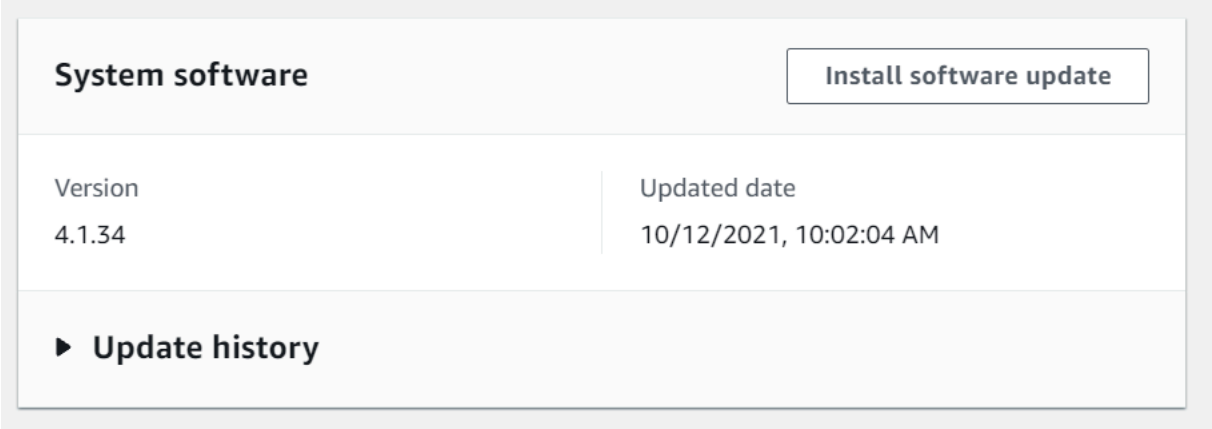
11. Seleziona Done (Fatto).

Aggiornare il software dell'appliance

L'appliance AWS Panorama dispone di diversi componenti software, tra cui un sistema operativo Linux, l'SDK per applicazioni AWS Panorama e librerie e framework di supporto per la visione artificiale. Per assicurarti di poter utilizzare le funzionalità e le applicazioni più recenti con il tuo dispositivo, aggiorna il software dopo la configurazione e ogni volta che è disponibile un aggiornamento.

Per aggiornare il software dell'appliance

1. Apri la pagina Dispositivi della console AWS Panorama.
2. Scegliete un elettrodomestico.
3. Scegli Impostazioni
4. In Software di sistema, scegli Installa aggiornamento software.



5. Scegli una nuova versione, quindi scegli Installa.

> **Important**
>
> Prima di continuare, rimuovi l'unità USB dall'accessorio e formattala per eliminarne il contenuto. L'archivio di configurazione contiene dati sensibili e non viene eliminato automaticamente.

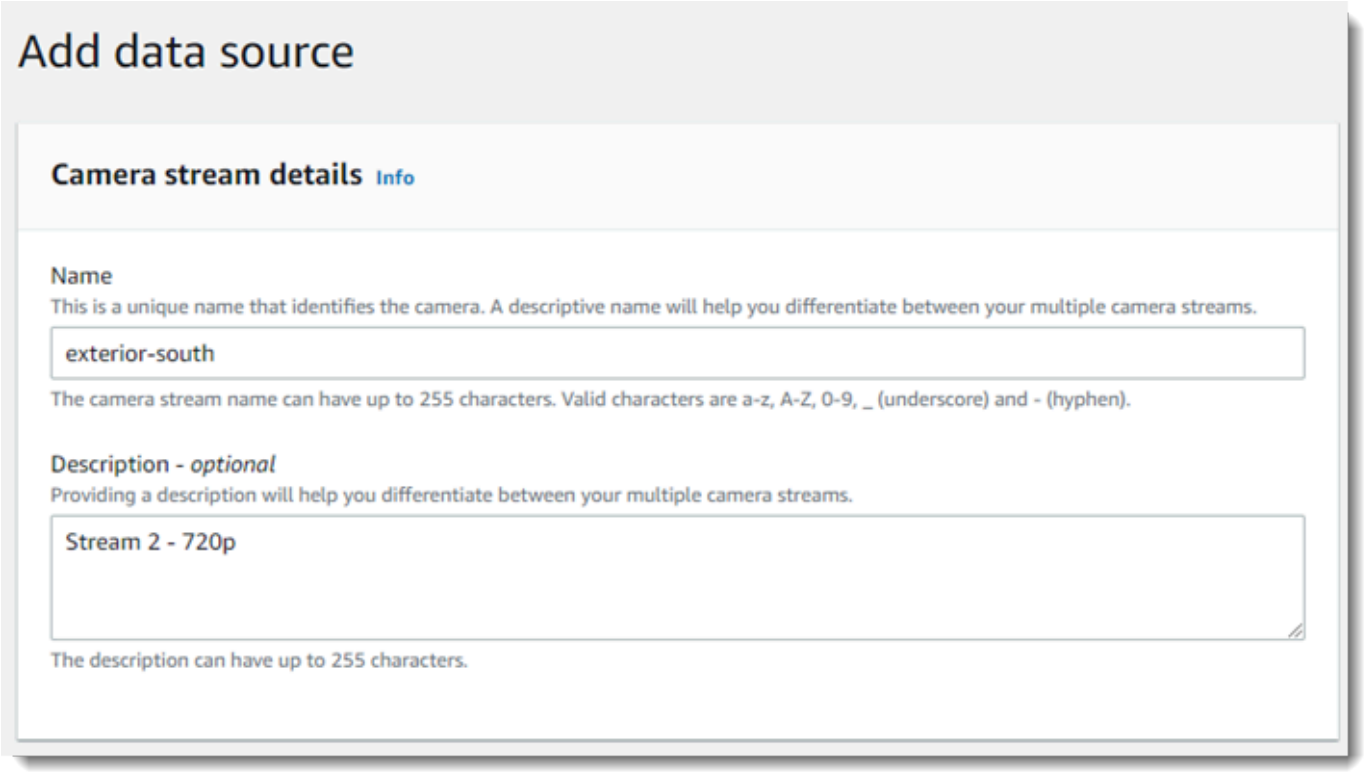
Il processo di aggiornamento può richiedere 30 minuti o più. Puoi monitorarne i progressi nella console AWS Panorama o su un monitor collegato. Al termine del processo, l'appliance si riavvia.

Aggiungi uno stream di videocamere

Successivamente, registra uno stream di videocamere con la console AWS Panorama.

Per registrare uno streaming di videocamere

1. Apri la pagina Sorgenti dati della console AWS Panorama.
2. Scegli Aggiungi origine dati.



3. Configura le impostazioni seguenti.

   - Nome: un nome per lo stream della videocamera.
   - Descrizione: una breve descrizione della fotocamera, della sua posizione o di altri dettagli.
   - URL RTSP: un URL che specifica l'indirizzo IP della telecamera e il percorso dello stream. Ad esempio, `rtsp://192.168.0.77/live/mpeg4/`
   - Credenziali: se lo streaming della videocamera è protetto da password, specificare il nome utente e la password.
4. Seleziona Salva.

AWS Panorama archivia le credenziali della videocamera in modo sicuro. AWS Secrets Manager Più applicazioni possono elaborare lo stesso flusso di telecamere contemporaneamente.

Fasi successive

Se hai riscontrato errori durante la configurazione, consultaRisoluzione dei problemi.

Per distribuire un'applicazione di esempio, vai all'argomento successivo.

Distribuzione dell'applicazione di esempio AWS Panorama

Dopo averconfigura la tua appliance AWS Panorama o un dispositivo compatibilee ha aggiornato il suo software, implementato un'applicazione di esempio. Nelle sezioni seguenti, importi un'applicazione di esempio con l'interfaccia a riga di comando dell'applicazione AWS Panorama e la distribuisci con la console AWS Panorama.

L'applicazione di esempio utilizza un modello di apprendimento automatico per classificare gli oggetti in fotogrammi video di una telecamera di rete. Utilizza l'AWS Panorama Application SDK per caricare un modello, ottenere immagini ed eseguire il modello. L'applicazione quindi sovrappone i risultati al video originale e lo invia a uno schermo collegato.

In un ambiente di vendita al dettaglio, l'analisi dei modelli di traffico pedonale consente di prevedere i livelli di traffico. Combinando l'analisi con altri dati, è possibile pianificare l'aumento delle esigenze di personale durante le festività e altri eventi, misurare l'efficacia delle pubblicità e delle promozioni di vendita o ottimizzare il posizionamento dei display e la gestione dell'inventario.

Sezioni

Prerequisiti

Per seguire le procedure in questa esercitazione, devi usare un terminale a riga di comando o una shell per eseguire i comandi. Negli elenchi di codice, i comandi sono preceduti da un simbolo di prompt ($) e dal nome della directory attuale, se appropriato.

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

Per comandi lunghi, utilizziamo un carattere di escape (\) per dividere un comando su più righe.

In Linux e macOS utilizzare la propria shell e il proprio programma di gestione dei pacchetti preferiti. In Windows 10 è possibile installare Windows Subsystem for Linux per ottenere una versione di Ubuntu e Bash integrata con Windows. Per informazioni sulla configurazione di un ambiente di sviluppo in Windows, vediConfigurazione di un ambiente di sviluppo in Windows.

Usi Python per sviluppare applicazioni AWS Panorama e installare strumenti con pip, il gestore di pacchetti di Python. Se non disponi già di Python,installare la versione più recente. Se hai Python 3 ma non pip, installa pip con il gestore di pacchetti del tuo sistema operativo o installa una nuova versione di Python, che viene fornita con pip.

In questo tutorial, usi Docker per creare il contenitore che esegue il codice dell'applicazione. Installa Docker dal sito Web di Docker: Ottieni Docker

Questo tutorial utilizza l'interfaccia a riga di comando dell'applicazione AWS Panorama per importare l'applicazione di esempio, creare pacchetti e caricare artefatti. L'interfaccia a riga di comando dell'applicazione AWS Panorama utilizzaAWS Command Line Interface(AWS CLI) per chiamare le operazioni API del servizio. Se hai già ilAWS CLI, aggiornarlo alla versione più recente. Per installare la CLI dell'applicazione AWS Panorama eAWS CLI, usopip.

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

Scarica l'applicazione di esempio ed estraila nel tuo spazio di lavoro.

- Applicazione di esempio–aws-panorama-sample.zip

Importazione dell'applicazione di esempio

Per importare l'applicazione di esempio da utilizzare nel tuo account, usa l'interfaccia a riga di comando dell'applicazione AWS Panorama. Le cartelle e il manifesto dell'applicazione contengono riferimenti a un numero di account segnaposto. Per aggiornarli con il tuo numero di account, esegui il`panorama-cli import-application`comando.

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

Il`SAMPLE_CODE`pacchetto, nel`packages`directory, contiene il codice e la configurazione dell'applicazione, incluso un Dockerfile che utilizza l'immagine di base dell'applicazione,`panorama-application`. Per creare il contenitore di applicazioni che viene eseguito sull'appliance, utilizzare il`panorama-cli build-container`comando.

```
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
```

Il passaggio finale con l'interfaccia a riga di comando dell'applicazione AWS Panorama consiste nel registrare il codice e i nodi del modello dell'applicazione e caricare le risorse su un punto di accesso Amazon S3 fornito dal servizio. Le risorse includono l'immagine del contenitore del codice, il modello e un file descrittore per ciascuna di esse. Per registrare i nodi e caricare le risorse, esegui il`panorama-cli package-application`comando.

```
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
```

Distribuzione dell'applicazione

Usa la console AWS Panorama per distribuire l'applicazione sul tuo dispositivo.

Per distribuire un'applicazione

1. Aprire la console AWS PanoramaPagina delle applicazioni distribuite.
2. ScegliDistribuzione dell'applicazione.
3. Incolla il contenuto del manifesto dell'applicazione,`graphs/aws-panorama-sample/graph.json`, nell'editor di testo. Seleziona Successivo.
4. Per Application name (Nome applicazione), immettere `aws-panorama-sample`.
5. ScegliContinua con la distribuzione.
6. ScegliInizia la distribuzione.
7. ScegliSuccessivosenza selezionare un ruolo.
8. ScegliSeleziona dispositivo, quindi scegli il tuo elettrodomestico. Seleziona Successivo.
9. SulSelezionare le origini datipasso, scegliVisualizza input (s)e aggiungi lo stream della tua videocamera come fonte di dati. Seleziona Successivo.

10. SulConfigurapasso, scegliSuccessivo.
11. ScegliDistribuzione, quindi selezionareFatto.
12. Nell'elenco delle applicazioni distribuite, scegliaws-panorama-sample.

Aggiorna questa pagina per gli aggiornamenti o utilizza il seguente script per monitorare la distribuzione dalla riga di comando.

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": {}
```

```
    }
]
...
```

Se l'applicazione non si avvia, controlla ilregistri delle applicazioni e dei dispositiviin Amazon CloudWatch Registri.

Visualizza l'output

Al termine della distribuzione, l'applicazione avvia l'elaborazione del flusso video e invia i registri a CloudWatch.

Per visualizzare i log in CloudWatch Log

1. Aprire ilPagina dei gruppi di log di CloudWatch Console Logs.
2. Trova i log delle applicazioni e delle appliance AWS Panorama nei seguenti gruppi:

   - Registri dei dispositivi–`/aws/panorama/devices/`*`device-id`*
   - Registri delle applicazioni–`/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
```

Per visualizzare l'uscita video dell'applicazione, collegare l'apparecchio a un monitor con un cavo HDMI. Per impostazione predefinita, l'applicazione mostra qualsiasi risultato di classificazione con una confidenza superiore al 20%.

Example squeezenet_classes.json

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

Il modello di esempio ha 1000 classi che includono molti animali, cibo e oggetti comuni. Prova a puntare la fotocamera verso una tastiera o una tazza da caffè.

Per semplicità, l'applicazione di esempio utilizza un modello di classificazione leggero. Il modello emette un singolo array con una probabilità per ciascuna delle sue classi. Le applicazioni del mondo reale utilizzano più frequentemente modelli di rilevamento di oggetti con output multidimensionali. Per applicazioni di esempio con modelli più complessi, vedereApplicazioni, script e modelli di esempio.

Abilitare l'SDK per Python

L'applicazione di esempio utilizzaAWS SDK for Python (Boto)per inviare metriche ad Amazon CloudWatch. Per abilitare questa funzionalità, crea un ruolo che conceda all'applicazione il permesso di inviare metriche e ridistribuisci l'applicazione con il ruolo associato.

L'applicazione di esempio vale anche perAWS CloudFormationmodello che crea un ruolo con le autorizzazioni necessarie. Per creare il ruolo, utilizzare`aws cloudformation deploy`comando.

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

Per ridistribuire l'applicazione

1. Aprire la console AWS PanoramaPagina delle applicazioni distribuite.
2. Scegliere un'applicazione.
3. Scegliere Replace (Sostituisci).
4. Completare i passaggi per implementare l'applicazione. NelSpecificare il ruolo IAM, scegli il ruolo appena creato. Il suo nome inizia con `aws-panorama-sample-runtime`.
5. Al termine della distribuzione, apri ilCloudWatchplanciae visualizzare i parametri nella`AWSPanoramaApplication`Spazio dei nomi. Ogni 150 frame, l'applicazione registra e carica le metriche per l'elaborazione dei frame e il tempo di inferenza.

Elimina

Se hai finito di lavorare con l'applicazione di esempio, puoi usare la console AWS Panorama per rimuoverla dall'appliance.

Per rimuovere l'applicazione dal dispositivo

1. Aprire la console AWS PanoramaPagina delle applicazioni distribuite.
2. Scegliere un'applicazione.

3. ScegliEliminazione dal dispositivo.

Fasi successive

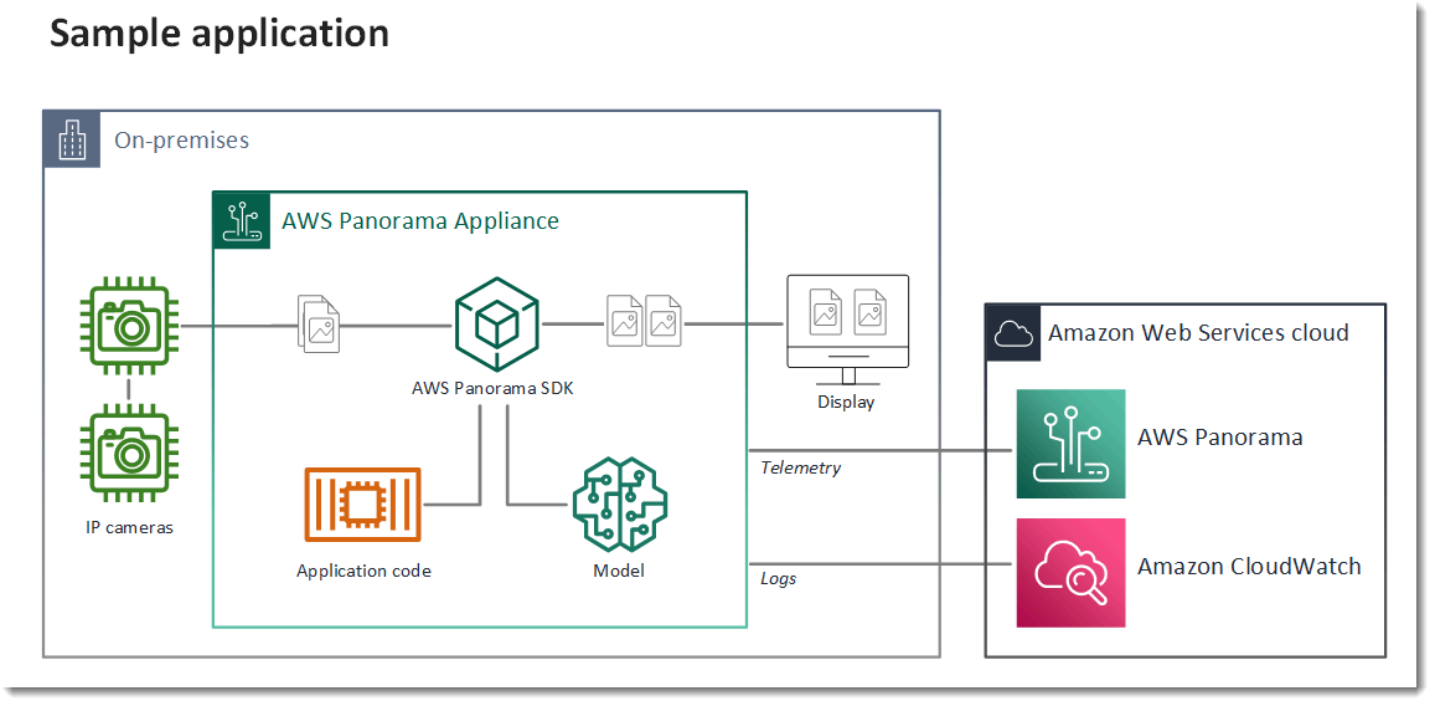
Se si sono verificati errori durante la distribuzione o l'esecuzione dell'applicazione di esempio, vedereRisoluzione dei problemi.

Per ulteriori informazioni sulle funzionalità e sull'implementazione dell'applicazione di esempio, continua al'argomento successivo.

Sviluppo di applicazioni AWS Panorama

È possibile utilizzare l'applicazione di esempio per imparare a conoscere la struttura dell'applicazione AWS Panorama e come punto di partenza per la tua applicazione.

Il seguente diagramma mostra i componenti principali dell'applicazione in esecuzione su AWS Panorama Appliance. Il codice dell'applicazione utilizza l'SDK dell'applicazione AWS Panorama per ottenere immagini e interagire con il modello, a cui non ha accesso diretto. L'applicazione invia video a uno schermo collegato ma non invia dati immagine al di fuori della rete locale.



In questo esempio, l'applicazione utilizza l'SDK dell'applicazione AWS Panorama per ottenere fotogrammi di video da una telecamera, preelaborare i dati video e inviare i dati a un modello di visione artificiale che rileva gli oggetti. L'applicazione visualizza il risultato su un display HDMI collegato all'apparecchio.

Sezioni

- Fasi successive

Manifest dell'applicazione

Il manifesto dell'applicazione è un file denominato`graph.json`nella`graphs`folder. Il manifesto definisce i componenti dell'applicazione, che sono pacchetti, nodi e bordi.

I pacchetti sono codice, file di configurazione e binari per il codice dell'applicazione, i modelli, le fotocamere e i display. L'applicazione PHP PHP di esempio utilizza 4 pacchetti:

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

```
    "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"
        }
    ],
```

I primi due pacchetti sono definiti all'interno dell'applicazione, nella`packages`directory. Contengono il codice e il modello specifici di questa applicazione. I secondi due pacchetti sono pacchetti generici di telecamere e display forniti dal servizio AWS Panorama. La`abstract_rtsp_media_source`package è un segnaposto per una fotocamera che si sostituisce durante la distribuzione. La`hdmi_data_sink`package rappresenta il connettore di uscita HDMI sul dispositivo.

I nodi sono interfacce per i pacchetti, così come parametri non di pacchetto che possono avere valori predefiniti che vengono sovrascritti al momento della distribuzione. I pacchetti di codice e modello

definiscono le interfacce in`package.json`file che specificano ingressi e uscite, che possono essere flussi video o un tipo di dati di base come float, booleano o stringa.

Ad esempio, le ricette`code_node`nodo si riferisce a un'interfaccia dal`SAMPLE_CODE`pacchetto.

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

Questa interfaccia è definita nel file di configurazione del pacchetto,`package.json`. L'interfaccia specifica che il pacchetto è business logic e che richiede un flusso video denominato`video_in`e un numero in virgola mobile denominato`threshold`come ingressi. L'interfaccia specifica inoltre che il codice richiede un buffer di flusso video denominato`video_out`per trasmettere video su uno schermo

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"
                    }
                ]
            }
        ]
    }
}
```

Di nuovo nel manifesto dell'applicazione, il`camera_node`nodo rappresenta un flusso video proveniente da una telecamera. Include un decoratore che appare nella console quando si distribuisce l'applicazione, che richiede di scegliere un flusso della telecamera.

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

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

Un nodo parametro,`threshold_param`, definisce il parametro della soglia di confidenza utilizzato dal codice dell'applicazione. Il valore predefinito è 60 e può essere ignorato durante la distribuzione.

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

```
            {
                "name": "threshold_param",
                "interface": "float32",
                "value": 60.0,
                "overridable": true,
                "decorator": {
                    "title": "Confidence threshold",
```

```
                    "description": "The minimum confidence for a classification to be
recorded."
                }
            }
```

La sezione finale del manifesto dell'applicazione,`edges`, effettua connessioni tra i nodi. Il flusso video della telecamera e il parametro threshold si connettono all'ingresso del nodo di codice e l'uscita video dal nodo di codice si connette al display.

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

```
        "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"
            }
        ]
```

Creazione con l'applicazione di esempio

È possibile utilizzare l'applicazione di esempio come punto di partenza per la tua applicazione.

Il nome di ciascun pacchetto deve essere univoco nell'account. Se tu e un altro utente nel tuo account utilizzate entrambi un nome di pacchetto generico come`codeomodel`, potresti ottenere la versione sbagliata del pacchetto quando esegui la distribuzione. Modifica il nome del pacchetto di codice in uno che rappresenta la tua applicazione.

Per rinominare il pacchetto di codice

1. Rinominare la cartella dei pacchetti:`packages/123456789012-`*`SAMPLE_CODE`*`-1.0/`.
2. Aggiorna il nome del pacchetto nelle seguenti posizioni.

   - Manifest dell'applicazione–`graphs/aws-panorama-sample/graph.json`

- Configurazione di Package–`packages/123456789012-SAMPLE_CODE-1.0/package.json`
- Script di compilazione–`3-build-container.sh`

Per aggiornare il codice dell'applicazione

1. Modificare il codice dell'applicazione in`packages/123456789012-SAMPLE_CODE-1.0/src/application.py`.
2. Per compilare il container, esegui`3-build-container.sh`.

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

La CLI elimina automaticamente la vecchia risorsa contenitore da`assets`e aggiorna la configurazione del pacchetto.

3. Per caricare i pacchetti, esegui`4-package-application.py`.
4. Aprire la console AWS PanoramaPagina Applicazioni distribuite.
5. Scegliere un'applicazione.
6. Scegliere Replace (Sostituisci).
7. Completare i passaggi per implementare l'applicazione. Se necessario, è possibile apportare modifiche al manifesto dell'applicazione, ai flussi della telecamera o ai parametri.

Modifica del modello di visione artificiale

L'applicazione di esempio include un modello di visione artificiale. Per utilizzare il tuo modello, modifica la configurazione del nodo del modello e utilizza l'interfaccia a riga di comando dell'applicazione AWS Panorama per importarlo come risorsa.

L'esempio seguente utilizza un SSD MXNet ResNet50 modelli scaricabili da questa guida GitHub repo:ssd_512_resnet50_v1_voc.tar.gz

Per modificare il modello dell'applicazione di esempio

1. Rinominare la cartella dei pacchetti in base al modello. Ad esempio, per`packages/`*`123456789012`*`-`*`SSD_512_RESNET50_V1_VOC`*`-1.0/`.
2. Aggiorna il nome del pacchetto nelle seguenti posizioni.

   - Manifest dell'applicazione–`graphs/aws-panorama-sample/graph.json`
   - Configurazione di Package–`packages/`*`123456789012`*`-`*`SSD_512_RESNET50_V1_VOC`*`-1.0/package.json`
3. Nel file di configurazione del pacchetto (`package.json`). Modifica il file.`assets`valore a un array vuoto.

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

4. Apri il file descrittore del pacchetto (`descriptor.json`). Aggiornamento di`frameworkeshapevalori` corrispondenti al tuo modello.

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

   Il valore performa,`1,3,512,512`, indica il numero di immagini che il modello prende come input (1), il numero di canali in ciascuna immagine (3: rosso, verde e blu) e le dimensioni dell'immagine (512 x 512). I valori e l'ordine dell'array variano a seconda dei modelli.

5. Importa il modello con l'interfaccia a riga di comando dell'applicazione AWS Panorama. L'interfaccia a riga di comando dell'applicazione AWS Panorama copia i file del modello e del descrittore nel`assets`cartella con nomi univoci e aggiorna la configurazione del pacchetto.

```
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. Per caricare il modello, eseguipanorama-cli package-application.

```
$ panorama-cli package-application
Uploading package SAMPLE_CODE
Patch Version 1844d5a59150d33f6054b04bac527a1771fd2365e05f990ccd8444a5ab775809
 already registered, ignoring upload
Uploading package SSD_512_RESNET50_V1_VOC
Patch version for the package
 244a63c74d01e082ad012ebf21e67eef5d81ce0de4d6ad1ae2b69d0bc498c8fd
upload: assets/
b1a1589afe449b346ff47375c284a1998c3e1522b418a7be8910414911784ce1.tar.gz to
 s3://arn:aws:s3:us-west-2:454554846382:accesspoint/panorama-123456789012-
wc66m5eishf4si4sz5jefhx
63a/123456789012/nodePackages/SSD_512_RESNET50_V1_VOC/binaries/
b1a1589afe449b346ff47375c284a1998c3e1522b418a7be8910414911784ce1.tar.gz
upload: assets/
a6a9508953f393f182f05f8beaa86b83325f4a535a5928580273e7fe26f79e78.json to
 s3://arn:aws:s3:us-west-2:454554846382:accesspoint/panorama-123456789012-
wc66m5eishf4si4sz5jefhx63
a/123456789012/nodePackages/SSD_512_RESNET50_V1_VOC/binaries/
a6a9508953f393f182f05f8beaa86b83325f4a535a5928580273e7fe26f79e78.json
{
    "ETag": "\"2381dabba34f4bc0100c478e67e9ab5e\"",
    "ServerSideEncryption": "AES256",
    "VersionId": "KbY5fpESdpYamjWZ0YyGqHo3.LQQWUC2"
}
Registered SSD_512_RESNET50_V1_VOC with patch version
 244a63c74d01e082ad012ebf21e67eef5d81ce0de4d6ad1ae2b69d0bc498c8fd
Uploading package SQUEEZENET_PYTORCH_V1
Patch Version 568138c430e0345061bb36f05a04a1458ac834cd6f93bf18fdacdffb62685530
 already registered, ignoring upload
```

7. Aggiornamento del codice dell'applicazione. La maggior parte del codice può essere riutilizzata. Il codice specifico per la risposta del modello è nellaprocess_resultsMetodo.

```
    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)
```

In base al modello in uso, potrebbe anche essere necessario aggiornare il`preprocess`Metodo.

Pre-elaborazione di immagini

Prima che l'applicazione invii un'immagine al modello, la prepara per l'inferenza ridimensionandola e normalizzando i dati colore. Il modello utilizzato dall'applicazione richiede un'immagine di 224 x 224 pixel con tre canali di colore, per corrispondere al numero di input nel suo primo livello. L'applicazione regola ogni valore di colore convertendolo in un numero compreso tra 0 e 1, sottraendo il valore medio per quel colore e dividendo per la deviazione standard. Infine, combina i canali di colore e li converte in un NumPy array che il modello può elaborare.

Example application.py— Pre-elaborazione

```
    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)
```

Questo processo fornisce i valori del modello in un intervallo prevedibile centrato attorno a 0. Corrisponde alla pre-elaborazione applicata alle immagini nel set di dati di addestramento, che è un approccio standard ma può variare in base al modello.

Caricamento di metriche con SDK per Python

L'applicazione PHP PHP PHP PHP di esempio utilizza SDK per Python per caricare le metriche su Amazon. CloudWatch.

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

Ottiene l'autorizzazione da un ruolo di runtime assegnato durante la distribuzione. Il ruolo è definito nella`aws-panorama-sample.yml` AWS CloudFormationModello.

Example aws-panorama-sample.yml

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

```
      Path: /service-role/
```

L'applicazione di esempio installa l'SDK per Python e altre dipendenze con pip. Quando si crea il contenitore dell'applicazione, il`Dockerfile`esegue comandi per installare le librerie in cima a ciò che viene fornito con l'immagine di base.

Example Dockerfile

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

Per utilizzare il pluginAWSSDK nel codice dell'applicazione, prima modifica il modello per aggiungere le autorizzazioni per tutte le azioni API utilizzate dall'applicazione. Aggiornamento diAWS CloudFormationstack eseguendo il`1-create-role.s`hogni volta che apporti una modifica. Quindi, implementa le modifiche al codice dell'applicazione.

Per le azioni che modificano o utilizzano risorse esistenti, è consigliabile ridurre al minimo l'ambito di questo criterio specificando un nome o un modello per la destinazione.`Resource`in una dichiarazione separata. Per i dettagli sulle azioni e le risorse supportate da ciascun servizio, vedere.Operazioni, risorse e chiavi di condizionenella Service Authorization Reference

Fasi successive

Per istruzioni sull'utilizzo dell'interfaccia a riga di comando di AWS Panorama Application per creare applicazioni e creare pacchetti da zero, consulta il README dell'interfaccia della riga di comando.

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

Per ulteriori esempi di codice e un'utilità di test da utilizzare per convalidare il codice dell'applicazione prima della distribuzione, visita il repository di campioni di AWS Panorama.

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

Modelli e fotocamere di visione artificiale supportati

AWS Panorama supporta modelli creati conPyTorch, Apache MXNet e. TensorFlow Quando distribuisci un'applicazione, AWS Panorama compila il tuo modello in SageMaker Neo. Puoi creare modelli in Amazon SageMaker o nel tuo ambiente di sviluppo, purché utilizzi livelli compatibili con SageMaker Neo.

Per elaborare video e ottenere immagini da inviare a un modello, l'appliance AWS Panorama si collega a un flusso video codificato H.264 con il protocollo RTSP. AWS Panorama verifica la compatibilità di una varietà di fotocamere comuni.

Sezioni

Modelli supportati

Quando crei un'applicazione per AWS Panorama, fornisci un modello di machine learning che l'applicazione utilizza per la visione artificiale. Puoi utilizzare modelli predefiniti e preaddestrati forniti da framework di modelli, un modello di esempio o un modello che crei e addestra tu stesso.

> **Note**
>
> Per un elenco di modelli predefiniti che sono stati testati con AWS Panorama, consulta Compatibilità dei modelli.

Quando distribuisci un'applicazione, AWS Panorama utilizza il compilatore SageMaker Neo per compilare il tuo modello di visione artificiale. SageMakerNeo è un compilatore che ottimizza i modelli per funzionare in modo efficiente su una piattaforma di destinazione, che può essere un'istanza in Amazon Elastic Compute Cloud (Amazon EC2) o un dispositivo edge come AWS Panorama Appliance.

AWS Panorama supporta le versioni di PyTorch Apache MXNet e TensorFlow che sono supportate per i dispositivi edge da SageMaker Neo. Quando crei il tuo modello, puoi utilizzare le versioni del framework elencate nelle note di rilascio di SageMaker Neo. NelSageMaker, puoi utilizzare l'algoritmo di classificazione delle immagini integrato.

Per ulteriori informazioni sull'utilizzo dei modelli in AWS Panorama, consultaModelli di visione artificiale.

Fotocamere supportate

L'appliance AWS Panorama supporta flussi video H.264 da telecamere che emettono RTSP su una rete locale. Per lo streaming della fotocamera superiore a 2 megapixel, l'appliance ridimensiona l'immagine a 1920x1080 pixel o una dimensione equivalente che preserva le proporzioni dello stream.

I seguenti modelli di fotocamera sono stati testati per verificarne la compatibilità con l'appliance AWS Panorama:

- Asse: M3057-PLVE, M3058-PLVE, P1448-LE, P3225-LV Mk II
- LaView— LV-PB3040W
- Vivotek — IB9360-H
- Amcrest — IP2M-841B
- Informazioni — IPC-B850W-S-3X, IPC-E250W-S
- WGCC — Dome PoE da 4 MP ONVIF

Per le specifiche hardware dell'appliance, vedereSpecifiche dell'appliance AWS Panorama.

Specifiche dell'appliance AWS Panorama

AWS Panorama Appliance ha le seguenti specifiche hardware. Per altridispositivi compatibili, consulta la documentazione del produttore.

Componente	Specifica
Processore e GPU	Nvidia Jetson AGX Xaviercon 32 GB di RAM
Ethernet	2x 1000 Base-T (Gigabyte)
USB	1 USB 2.0 e 1 USB 3.0 tipo A femmina
Uscita HDMI	2.0a
Dimensioni	7,75 «x 9,6 «x 1,6» (197mm x 243mm x 40 mm)
Weight	1,7 kg (3,7 libbre)
Alimentatore	100V-240V 50-60Hz AC 65 W
Ingresso energetico	Presa IEC 60320 C6 (3 pin)
Protezione da polvere e liquidi	IP-62
Conformità alle normative EMI/EMC	FCC Part-15 (USA)
Limiti termici touch	IEC-62368
Temperatura operativa	Da -20°C a 60°C
Umidità operativa	Da 0% a 95% RH
Temperatura di stoccaggio	Da -20°C a 85°C
Storage dell'umidità	Non controllato per basse temperature. 90% RH ad alta temperatura
raffreddamento	Estrazione di calore ad aria forzata (ventola)

Componente	Specifica
Opzioni di montaggio	Montaggio su rack o autoportante
Cavo di alimentazione	1,8 metri
Controllo energetico	Pulsante
Reimposta	Switch momentaneo
LED di stato e di rete	LED RGB programmabile a 3 colori

L'unità di archiviazione Wi-Fi, Bluetooth e scheda SD sono presenti sull'apparecchiatura ma non sono utilizzabili.

AWS Panorama Appliance include due viti per il montaggio su un server rack. È possibile montare due apparecchi side-by-side su un rack da 19 pollici.

Service Quotas

AWS Panorama applica delle quote alle risorse che crei nel tuo account e alle applicazioni che distribuisci. Se utilizzi AWS Panorama in piùAWSRegioni, le quote si applicano separatamente a ciascuna regione. Le quote di AWS Panorama non sono regolabili.

Le risorse in AWS Panorama includono dispositivi, pacchetti di nodi applicativi e istanze di applicazioni.

- Dispositivi— Fino a 50 dispositivi registrati per regione.
- Pacchetti di nodi— 50 pacchetti per regione, con un massimo di 20 versioni per pacchetto.
- Istanze dell'applicazione— Fino a 10 applicazioni per dispositivo. Ogni applicazione può monitorare fino a 8 stream di telecamere. Le distribuzioni sono limitate a 200 al giorno per ogni dispositivo.

Quando utilizzi l'interfaccia a riga di comando dell'applicazione AWS Panorama,AWS Command Line Interface, oppureAWSSDK con il servizio AWS Panorama, le quote si applicano al numero di chiamate API effettuate. Puoi effettuare fino a 5 richieste in totale al secondo. Un sottoinsieme di operazioni API che creano o modificano risorse applica un limite aggiuntivo di 1 richiesta al secondo.

Per un elenco completo delle quote, visita ilConsole Service Quotas, oppure vediEndpoint e quote di AWS PanoramanelRiferimenti generali di Amazon Web Services.

Autorizzazioni AWS Panorama

È possibile usareAWS Identity and Access Management (IAM) per gestire l'accesso alAWS Panorama servizio e alle risorse, ad esempio appliance e applicazioni. Per gli utenti del proprio account che utilizzanoAWS Panorama, è necessario gestire le autorizzazioni in una policy di autorizzazioni applicabile ai ruoli IAM. Per gestire le autorizzazioni per un'applicazione, è necessario creare un ruolo e assegnarlo all'applicazione.

Per gestire le autorizzazioni per gli utenti nel proprio account, utilizza la policy gestita cheAWS Panorama fornisce, oppure scrivile tu stesso. Hai bisogno delle autorizzazioni per accedere ad altriAWS servizi per ottenere i log delle applicazioni e delle appliance, visualizzare le metriche e assegnare un ruolo a un'applicazione.

Un'AWS Panoramaappliance inoltre ha un ruolo che consente di concedere l'autorizzazione per accedere aiAWS servizi e alle risorse. Il ruolo dell'appliance è uno dei ruoli di servizio che ilAWS Panorama servizio utilizza per accedere ad altri servizi per conto dell'utente.

Un ruolo dell'applicazione è un ruolo di servizio separato creato per un'applicazione, per concederle l'autorizzazione a utilizzareAWS i servizi conAWS SDK for Python (Boto). Per creare un ruolo dell'applicazione, sono necessari i privilegi amministrativi o l'aiuto di un amministratore.

È possibile limitare le autorizzazioni utente in base alla risorsa su cui influisce un'azione e, in alcuni casi, in base a condizioni aggiuntive. Ad esempio, è possibile specificare un motivo per l'Amazon Resource Name (ARN) di un'applicazione che richiede all'utente di includere il proprio nome utente nel nome delle applicazioni che creano. Per le risorse e le condizioni supportate da ciascuna operazione, consulta Operazioni, risorse e chiavi di condizioneAWS Panorama in Riferimenti alle autorizzazioni del servizio.

Per ulteriori informazioni, consulta la pagina Che cos'è IAM? nella Guida per l'utente di IAM.

Argomenti

Policy IAM basate su identità per AWS Panorama

Per concedere l'accesso agli utenti all'account AWS Panorama, usare le policy basate su identità inAWS Identity and Access Management (IAM). Applica policy basate su identità ai ruoli IAM associati a un utente. È possibile concedere le autorizzazioni a utenti in modo che assumano un ruolo nel proprio account ed eseguano l'accesso alle risorse AWS Panorama.

AWS Panorama offre policy gestite che concedono l'accesso alle azioni API AWS e, in alcuni casi, l'accesso ad altri servizi utilizzati per sviluppare e gestire le risorse AWS Panorama. AWS Panorama aggiorna le policy gestite secondo le necessità, per assicurare che gli utenti possano accedere a nuove caratteristiche quando vengono rilasciate.

- AWSPanoramaFullAccess— Fornisce accesso completo ad AWS Panorama, ai punti di accesso AWS Panorama in Amazon S3, alle credenziali dell'appliance e ai log delle appliance in Amazon CloudWatch.AWS Secrets Manager Include l'autorizzazione a creare un ruolo collegato ai servizi per AWS Panorama. Visualizza la politica

La`AWSPanoramaFullAccess` policy consente di etichettare le risorse AWS Panorama, ma non dispone di tutte le autorizzazioni relative ai tag utilizzate dalla console AWS Panorama. Per concedere le autorizzazioni.

- ResourceGroupsandTagEditorFullAccess— Visualizza la politica

La`AWSPanoramaFullAccess` politica non include l'autorizzazione all'acquisto di dispositivi dalla console AWS Panorama. Per concedere le autorizzazioni.

- ElementalAppliancesSoftwareFullAccess— Visualizza la politica

Le policy gestite concedono le autorizzazioni alle operazioni API senza limitare le risorse che un utente può modificare. Per un controllo ancora più accurato, è possibile creare le proprie policy che limitano l'ambito di applicazione delle autorizzazioni di un utente. Usa la politica di accesso completo come punto di partenza per le tue politiche.

> **Creazione di ruoli di servizio**
>
> La prima volta che usi la console AWS Panorama, devi avere l'autorizzazione per creare il ruolo di servizio utilizzato dall'appliance AWS Panorama. Un ruolo di servizio fornisce a un

servizio l'autorizzazione a gestire le risorse o interagire con altri servizi. Crea questo ruolo prima di concedere l'accesso ai tuoi utenti.

Per dettagli sulle risorse e sulle condizioni che puoi utilizzare per limitare l'ambito delle autorizzazioni di un utente in AWS Panorama, consulta Azioni, risorse e chiavi di condizione per AWS Panorama nel Service Authorization Reference.

Ruoli di servizio AWS Panorama e risorse interservizi

AWS Panorama utilizza altri servizi AWS per gestire l'appliance AWS Panorama, archiviare dati e importare risorse applicative. Un ruolo di servizio consente a un servizio di gestire le risorse o interagire con altri servizi. Quando accedi alla console AWS Panorama per la prima volta, crei i seguenti ruoli di servizio:

- AWSServiceRoleForAWSPanorama— Consente ad AWS Panorama di gestire le risorse in AWS IoT, AWS Secrets Manager e AWS Panorama.

  Policy gestita:AWSPanoramaServiceLinkedRolePolicy
- AWSPanoramaApplianceServiceRole— Consente a un'appliance AWS Panorama di caricare i log su CloudWatche per ottenere oggetti dai punti di accesso Amazon S3 creati da AWS Panorama.

  Policy gestita:AWSPanoramaApplianceServiceRolePolicy

Per visualizzare le autorizzazioni associate a ciascun ruolo, utilizzare ilConsole IAM. Ove possibile, le autorizzazioni del ruolo sono limitate alle risorse che corrispondono a uno schema di denominazione utilizzato da AWS Panorama. Ad esempio,`AWSServiceRoleForAWSPanorama`concede solo l'autorizzazione per l'accesso al servizioAWS IoTrisorse che hanno`panorama`a loro nome.

Sezioni

- Garantire il ruolo dell'appliance
- Uso di altri servizi

Garantire il ruolo dell'appliance

L'appliance AWS Panorama utilizza`AWSPanoramaApplianceServiceRole`ruolo per accedere alle risorse del tuo account. L'appliance è autorizzata per caricare i log in CloudWatch Registra, legge le credenziali dello streaming della telecamera daAWS Secrets Managere per accedere a artefatti dell'applicazione nei punti di accesso Amazon Simple Storage Service (Amazon S3) creati da AWS Panorama.

> **Note**
>
> Le applicazioni non utilizzano le autorizzazioni del dispositivo. Per concedere all'applicazione l'autorizzazione all'usoAWSservizi, crea unruolo dell'applicazione.

AWS Panorama utilizza lo stesso ruolo di servizio con tutte le appliance del tuo account e non utilizza ruoli tra più account. Per un ulteriore livello di sicurezza, è possibile modificare la politica di attendibilità del ruolo dell'appliance per applicarla in modo esplicito, una procedura consigliata quando si utilizzano i ruoli per concedere a un servizio l'autorizzazione ad accedere alle risorse del proprio account.

Per aggiornare la politica di attendibilità dei ruoli dell'

1. Apri il ruolo dell'appliance nella console IAM:AWSPanoramaApplianceServiceRole
2. Seleziona Edit trust relationship (Modifica relazione di trust).
3. Aggiorna il contenuto della politica e poi scegliAggiorna policy di trust.

La seguente politica di attendibilità include una condizione che garantisce che quando AWS Panorama assume il ruolo di appliance, lo faccia per un dispositivo incluso nel tuo account. Il`aws:SourceAccount`condition confronta l'ID dell'account specificato da AWS Panorama con quello incluso nella policy.

Example politica di fiducia — Account specifico

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

```
  ]
}
```

Se desideri limitare ulteriormente AWS Panorama e consentirgli di assumere il ruolo solo con un dispositivo specifico, puoi specificare il dispositivo tramite ARN. Il`aws:SourceArn`condition confronta l'ARN dell'appliance specificato da AWS Panorama con quello incluso nella policy.

Example policy di attendibilità: dispositivo singolo

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

Se si ripristina e si rifornisce il dispositivo, è necessario rimuovere temporaneamente la condizione ARN di origine e quindi aggiungerla nuovamente con il nuovo ID del dispositivo.

Per ulteriori informazioni su queste condizioni e su best practice di sicurezza quando i servizi utilizzano i ruoli per accedere a risorse nel proprio account, consultareProblema del "confused deputy"nella Guida per l'utente di IAM.

Uso di altri servizi

AWS Panorama crea o accede alle risorse nei seguenti servizi:

- AWS IoT— Cose, politiche, certificati e lavori per l'appliance AWS Panorama
- Amazon S3— Punti di accesso per la gestione temporanea di modelli, codice e configurazioni delle applicazioni.
- Secrets Manager— Credenziali a breve termine per AWS Panorama Appliance.

Per informazioni sul formato Amazon Resource Name (ARN) o sugli ambiti di autorizzazione per ciascun servizio, consulta gli argomenti nellaIAM User Guideche sono collegati a questo elenco.

Concessione delle autorizzazioni a un'applicazione

È possibile creare un ruolo per la tua applicazione per concederle il permesso di chiamareAWSServizi . Per impostazione predefinita, le applicazioni non dispongono di autorizzazioni. È possibile creare un ruolo di applicazione in IAM e assegnarlo a un'applicazione durante la distribuzione. Per concedere alla tua applicazione solo le autorizzazioni necessarie, crea un ruolo con le autorizzazioni per azioni API specifiche.

LaApplicazione di esempioinclude unAWS CloudFormationmodello e script che creano un ruolo di applicazione. È unRuolo del servizioche AWS Panorama può supporre. Questo ruolo concede all'applicazione il permesso di chiamare CloudWatch per caricare le metriche.

Example aws-panorama-sample.yml— Ruolo dell'applicazione

```
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/
```

È possibile estendere questo script per concedere autorizzazioni ad altri servizi, specificando un elenco di azioni o modelli API per il valore di`Action`.

Per ulteriori informazioni sulle autorizzazioni in AWS Panorama, consulta.Autorizzazioni AWS Panorama.

Gestione diAWS PanoramaAppliance

LaAWS PanoramaL'appliance è l'hardware che esegue le applicazioni. Utilizzi il moduloAWS Panoramaconsole per registrare un'appliance, aggiornare il software e distribuirvi applicazioni. Il software sulAWS PanoramaL'appliance si collega ai flussi di telecamere, invia fotogrammi di video all'applicazione e visualizza l'uscita video su un display collegato.

Dopo aver configurato l'apparecchio o un altrodispositivo compatibile, si registrano telecamere da utilizzare con le applicazioni. Utente correntegestire i flussi di telecamerenellaAWS Panoramaconsole. Quando si distribuisce un'applicazione, si sceglie quale videocamera invia l'appliance per l'elaborazione.

Per esercitazioni che introducono ilAWS PanoramaApparecchio con applicazione campione, vedereNozioni di base su AWS Panorama.

Argomenti

- Gestione di un'appliance di AWS Panorama di AWS
- Connessione di AWS Panorama Appliance alla rete
- Gestione dei flussi di telecamere in AWS Panorama
- Gestisci applicazioni su un'appliance di AWS Panorama
- Pulsanti e luci di AWS Panorama

Gestione di un'appliance di AWS Panorama di AWS

Utilizzi la console AWS Panorama per configurare, aggiornare o annullare la registrazione dell'appliance AWS Panorama e di altri dispositivi compatibili.

Per configurare un dispositivo, segui le istruzioni nel tutorial introduttivo. Il processo di configurazione crea le risorse in AWS Panorama che tracciano l'appliance e coordinano gli aggiornamenti e le distribuzioni.

Per registrare un'appliance con l'API AWS Panorama, consultaAutomatizza la registrazione del dispositivo.

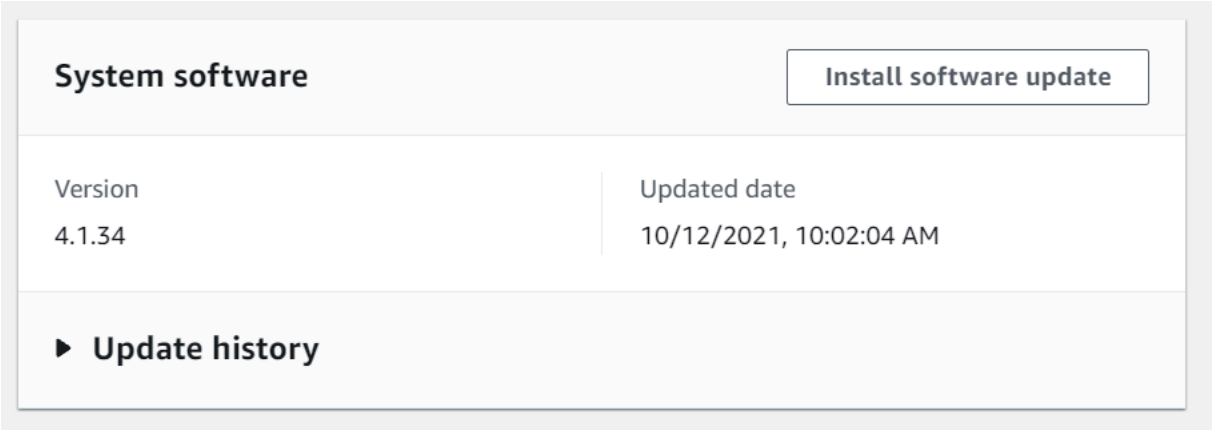
Sezioni

Aggiornare il software dell'appliance

Puoi visualizzare e distribuire gli aggiornamenti software per l'appliance nella console AWS Panorama. Gli aggiornamenti possono essere obbligatori o facoltativi. Quando è disponibile un aggiornamento necessario, la console chiede di applicarlo. È possibile applicare aggiornamenti opzionali nella pagina Impostazioni dell'appliance.

Per aggiornare il software dell'appliance

1. Apri la pagina dei dispositivi della console AWS Panorama.
2. Scegli un elettrodomestico.
3. Scegli Impostazioni
4. In Software di sistema, scegli Installa l'aggiornamento software.



5. Scegli una nuova versione e quindi scegli Installa.

Annullamento della registrazione di un'appliance

Se hai finito di lavorare con un'appliance, puoi utilizzare la console AWS Panorama per annullare la registrazione ed eliminare leAWS IoT risorse associate.

Per eliminare un dispositivo

1. Apri la pagina dei dispositivi della console AWS Panorama.
2. Scegliete il nome del dispositivo.
3. Scegli Delete (Elimina).
4. Immettete il nome del dispositivo e scegliete Elimina.

Quando si elimina un dispositivo dal servizio AWS Panorama, i dati sull'appliance non vengono eliminati automaticamente. Un dispositivo annullato non può connettersi aiAWS servizi e non può essere registrato nuovamente finché non viene ripristinato.

Riavvio di un'appliance

È possibile riavviare un dispositivo da remoto.

Per riavviare un dispositivo

1. Apri la pagina dei dispositivi della console AWS Panorama.
2. Scegliete il nome del dispositivo.
3. Scegliere Reboot (Riavvia).

La console invia un messaggio all'appliance per riavviarlo. Per ricevere il segnale, l'apparecchio deve essere in grado di connettersiAWS IoT. Per riavviare un'appliance con l'API AWS Panorama, vediRiavvio delle appliance.

Reimpostazione di un'appliance

Per utilizzare un'appliance in una regione diversa o con un account diverso, è necessario reimpostazione e riimpostazione con un nuovo certificato. La reimpostazione del dispositivo applica la versione software richiesta più recente ed elimina tutti i dati dell'account.

Per avviare un'operazione di ripristino, l'apparecchio deve essere collegato e spento. Tieni premuti entrambi i pulsanti di accensione e ripristino per cinque secondi. Quando si rilasciano i pulsanti, la spia di stato lampeggia in arancione. Attendere che la spia di stato lampeggi in verde prima di rifornire o scollegare l'apparecchiatura.

È inoltre possibile ripristinare il software dell'appliance senza eliminare i certificati dal dispositivo. Per ulteriori informazioni, consultare Pulsanti di accensione e ripristino.

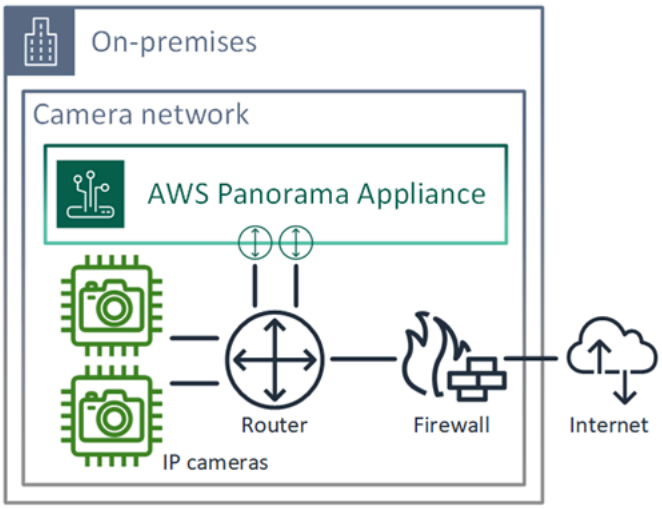
Connessione di AWS Panorama Appliance alla rete

L'AWS Panorama Appliance richiede la connettività sia al AWS cloud che alla rete locale di telecamere IP. Puoi connettere l'appliance a un singolo firewall che garantisce l'accesso a entrambi o connettere ciascuna delle due interfacce di rete del dispositivo a una sottorete diversa. In entrambi i casi, è necessario proteggere le connessioni di rete dell'appliance per impedire l'accesso non autorizzato ai flussi delle telecamere.

Sections

Configurazione di rete singola

L'appliance dispone di due porte Ethernet. Se si indirizza tutto il traffico da e verso il dispositivo attraverso un singolo router, è possibile utilizzare la seconda porta per la ridondanza in caso di interruzione della connessione fisica alla prima porta. Configurate il router per consentire all'appliance di connettersi solo ai flussi delle telecamere e a Internet e per impedire che i flussi delle telecamere lascino altrimenti la rete interna.

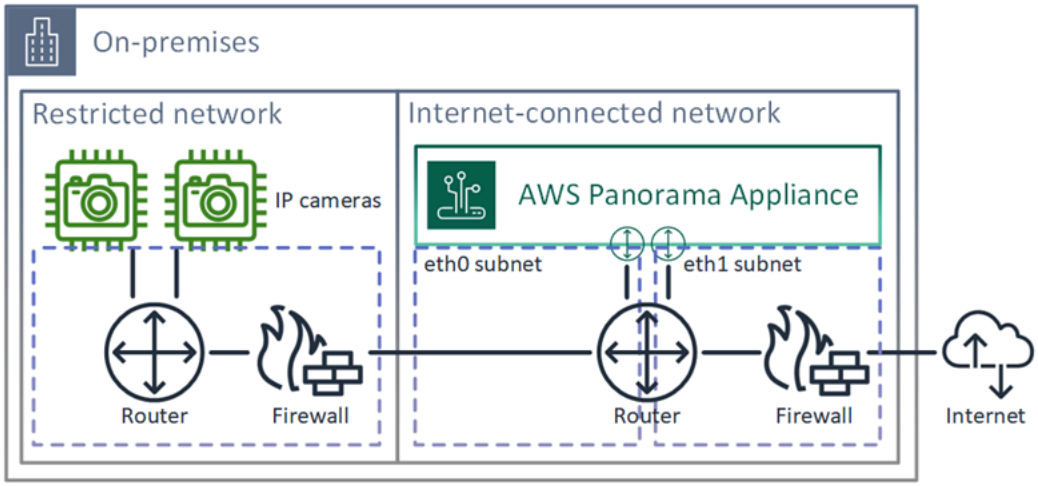


Per informazioni dettagliate sulle porte e sugli endpoint a cui l'appliance deve accedere, vedere e. Configurazione dell'accesso al servizio Configurazione dell'accesso alla rete locale

Configurazione a doppia rete

Per un ulteriore livello di sicurezza, è possibile collocare l'appliance in una rete connessa a Internet separata dalla rete di telecamere. Un firewall tra la rete di telecamere con restrizioni e la rete dell'appliance consente solo all'appliance di accedere ai flussi video. Se la rete di telecamere era precedentemente protetta per motivi di sicurezza, potreste preferire questo metodo piuttosto che collegare la rete di telecamere a un router che consente anche l'accesso a Internet.

L'esempio seguente mostra l'appliance che si connette a una sottorete diversa su ciascuna porta. Il router colloca l'`eth0`interfaccia su una sottorete che indirizza verso la rete di telecamere e `eth1` su una sottorete che indirizza verso Internet.



Puoi confermare l'indirizzo IP e l'indirizzo MAC di ogni porta nella console AWS Panorama.

Configurazione dell'accesso al servizio

Durante il provisioning, è possibile configurare l'appliance per richiedere un indirizzo IP specifico. Scegliete un indirizzo IP in anticipo per semplificare la configurazione del firewall e assicurarvi che l'indirizzo dell'appliance non cambi se rimane offline per un lungo periodo di tempo.

L'appliance utilizza i AWS servizi per coordinare gli aggiornamenti e le distribuzioni del software. Configura il firewall per consentire all'appliance di connettersi a questi endpoint.

Accesso a Internet

- AWS IoT(HTTPS e MQTT, porte 443, 8443 e 8883) e endpoint di gestione dei dispositivi. AWS IoT Core Per i dettagli, consulta gli endpoint e le quote di AWS IoT Device Management nel. Riferimenti generali di Amazon Web Services
- AWS IoTcredenziali (HTTPS, porta 443) e sottodomini. `credentials.iot.<region>.amazonaws.com`
- Amazon Elastic Container Registry (HTTPS, porta 443) d`kr.ecr.<region>.amazonaws.com` e sottodomini. `api.ecr.<region>.amazonaws.com`
- Amazon CloudWatch (HTTPS, porta 443) —`monitoring.<region>.amazonaws.com`.
- Amazon CloudWatch Logs (HTTPS, porta 443) —. `logs.<region>.amazonaws.com`
- Amazon Simple Storage Service (HTTPS, porta 443) `s3-accesspoint.<region>.amazonaws.com` e sottodomini. `s3.<region>.amazonaws.com`

Se l'applicazione chiama altri AWS servizi, l'appliance deve accedere agli endpoint anche per tali servizi. Per ulteriori informazioni, consulta Service endpoints and quotas.

Configurazione dell'accesso alla rete locale

L'appliance deve accedere ai flussi video RTSP localmente, ma non tramite Internet. Configura il firewall per consentire all'appliance di accedere ai flussi RTSP sulla porta 554 internamente e per impedire agli stream di uscire o entrare da Internet.

Accesso locale

- Protocollo di streaming in tempo reale (RTSP, porta 554): per leggere gli stream delle telecamere.
- Network Time Protocol (NTP, porta 123): per mantenere sincronizzato l'orologio dell'appliance. Se non si utilizza un server NTP sulla rete, l'appliance può anche connettersi a server NTP pubblici tramite Internet.

Connettività privata

L'AWS Panorama Appliance non necessita di accesso a Internet se viene distribuita in una sottorete VPC privata con una connessione VPN a. AWS Puoi utilizzare la VPN da sito a sito AWS Direct Connect o creare una connessione VPN tra un router locale e. AWS All'interno della tua sottorete VPC privata, crei endpoint che consentono all'appliance di connettersi ad Amazon Simple Storage

Service AWS IoT e ad altri servizi. Per ulteriori informazioni, consulta Connessione di un'appliance a una sottorete privata.

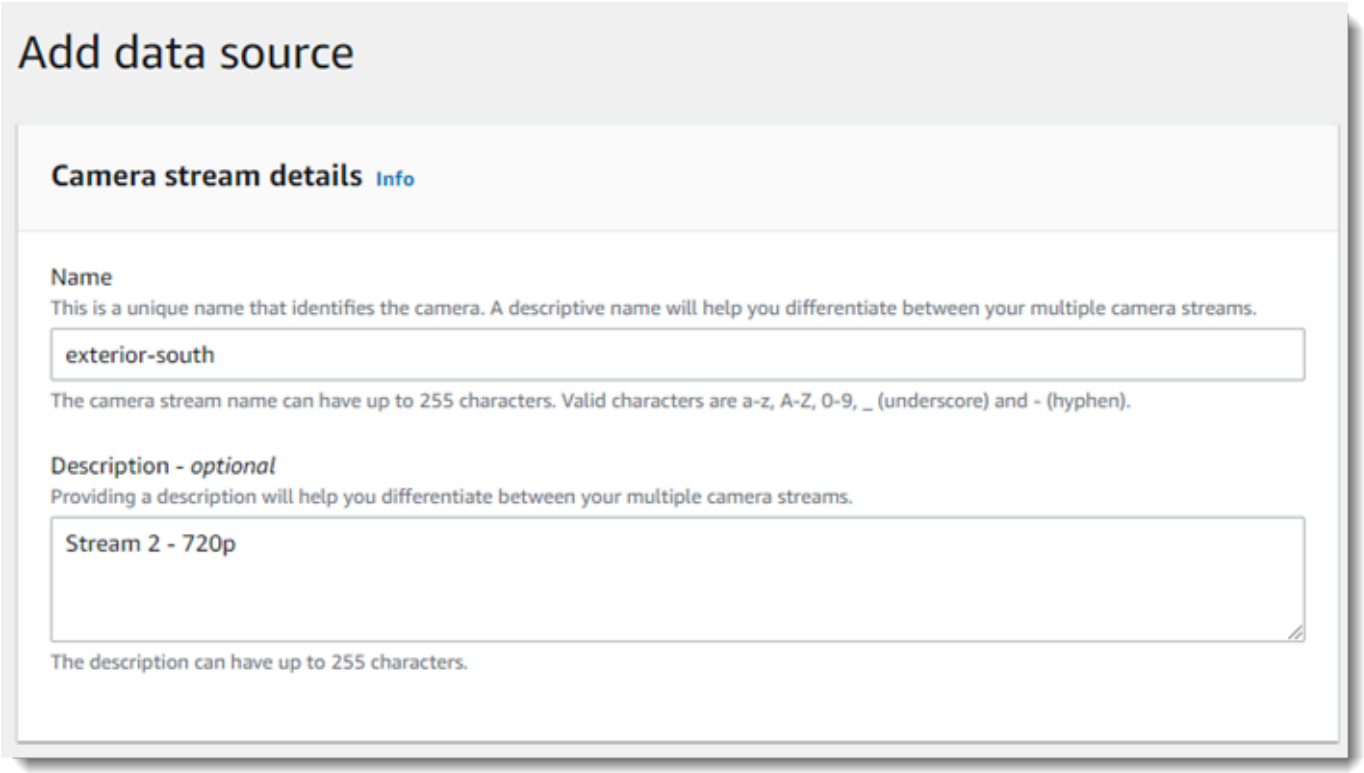
Gestione dei flussi di telecamere in AWS Panorama

Per registrare flussi video come origini dati per la tua applicazione, usa la console AWS Panorama. Un'applicazione può elaborare più flussi contemporaneamente e più dispositivi possono connettersi allo stesso flusso.

> **Important**
>
> Un'applicazione può connettersi a qualsiasi flusso di telecamere instradabile dalla rete locale a cui si connette. Per proteggere i tuoi flussi video, configura la tua rete in modo che consenta solo il traffico RTSP in locale. Per ulteriori informazioni, consulta la pagina Sicurezza AWS Panorama .

Per registrare un flusso di telecamere

1. Aprire la console AWS PanoramaOrigini dati.
2. ScegliereAggiungere origine dati.



3. Configura le impostazioni seguenti.

- Nome— Un nome per il flusso della telecamera.
- Description (Descrizione)— Una breve descrizione della fotocamera, della sua posizione o di altri dettagli.
- URL RTSP— URL che specifica l'indirizzo IP della telecamera e il percorso dello stream. Ad esempio, `rtsp://192.168.0.77/live/mpeg4/`
- Credenziali— Se il flusso della fotocamera è protetto con password, specificare il nome utente e la password.

4. Scegli Save (Salva).

Per registrare un flusso di telecamere con l'API AWS Panorama, consultaAutomatizza la registrazione del dispositivo.

Per un elenco di telecamere compatibili con AWS Panorama Appliance, vedereModelli e fotocamere di visione artificiale supportati.

Rimozione di un flusso

È possibile eliminare un flusso di telecamere nella console AWS Panorama.

Per rimuovere un flusso di telecamere

1. Aprire la console AWS PanoramaOrigini dati.
2. Scegli un flusso di telecamere.
3. ScegliereEliminare origine dati.

La rimozione di un flusso di telecamere dal servizio non interrompe l'esecuzione delle applicazioni o elimina le credenziali della videocamera da Secrets Manager. Per eliminare i segreti, usa ilConsole Secrets Manager.

Gestisci applicazioni su un'appliance di AWS Panorama

Un'applicazione è una combinazione di codice, modelli e configurazione. Dadispositivipagina nella console AWS Panorama, è possibile gestire le applicazioni sull'appliance.

Per gestire applicazioni su un'appliance di AWS Panorama

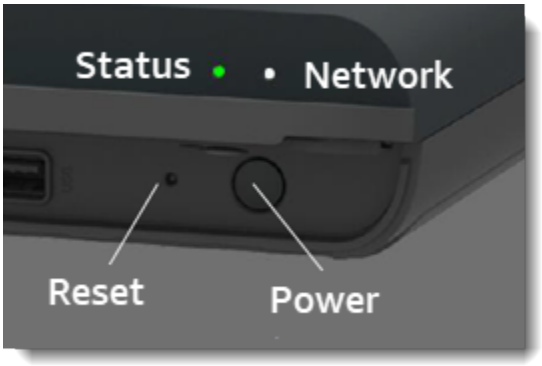
1. Aprire la console AWS PanoramaLa pagina dispositivi.
2. Scegli un apparecchio.

LaApplicazioni distribuitepagina mostra le applicazioni che sono state distribuite sull'appliance.

Utilizzare le opzioni in questa pagina per rimuovere le applicazioni distribuite dall'appliance o sostituire un'applicazione in esecuzione con una nuova versione. È inoltre possibile clonare un'applicazione (in esecuzione o eliminata) per distribuirne una nuova copia.

Pulsanti e luci di AWS Panorama

L'appliance AWS Panorama ha due luci LED sopra il pulsante di accensione che indicano lo stato del dispositivo e la connettività di rete.



Luce di stato

I LED cambiano colore e lampeggiano per indicare lo stato. Un lampeggiamento lento è una volta ogni tre secondi. Un lampeggio veloce è una volta al secondo.

Stati LED di stato

- Fast lamping verde— L'apparecchio si sta avviando.
- Verde solido— L'apparecchio funziona normalmente.
- Lampeggia lentamente— L'appliance sta copiando i file di configurazione e sta tentando di registrarsi conAWS IoT.
- Fast lamping blu— L'apparecchio ècopia di un'immagine di registrosu un'unità USB.
- Fast lamping rosso— L'apparecchio ha riscontrato un errore durante l'avvio o si è surriscaldato.
- Arancione che lampeggia lentamente— L'appliance sta ripristinando la versione software più recente.
- Fast lamping arancione— Il dispositivo sta ripristinando la versione minima del software.

Luce di rete

Il LED di rete ha i seguenti stati:

Stati LED di rete

- Verde solido— È collegato un cavo Ethernet.

- Verde lampeggiante— L'apparecchio comunica tramite la rete.
- Rosso solido— Un cavo Ethernet non è collegato.

Pulsanti di accensione e ripristino

I pulsanti di accensione e ripristino si trovano nella parte anteriore del dispositivo sotto una custodia protettiva. Il pulsante di ripristino è più piccolo e incassato. Usa un piccolo cacciavite o una graffetta per premerlo.

Per ripristinare un dispositivo

1. L'apparecchio deve essere collegato e spento. Per spegnere l'apparecchio, tieni premuto il pulsante di accensione per 1 secondo e attendi il completamento della sequenza di spegnimento. La sequenza di spegnimento richiede circa 10 secondi.
2. Per ripristinare l'apparecchio, utilizzare le seguenti combinazioni di pulsanti. Una pressione breve è di 1 secondo. Una pressione prolungata è 5 secondi. Per le operazioni che richiedono più pulsanti, tieni premuti entrambi i pulsanti contemporaneamente.

   - Ripristino completo— Premere a lungo per accendere e ripristinare.

     Ripristina la versione minima del software ed elimina tutti i file e le applicazioni di configurazione.
   - Ripristina l'ultima versione del software— Premere brevemente il reset.

     Riapplica l'aggiornamento software più recente al dispositivo.
   - Ripristina la versione minima del software— Premere a lungo il reset.

     Riapplica l'ultimo aggiornamento software richiesto al dispositivo.
3. Rilascia entrambi i pulsanti. L'apparecchio si accende e la spia di stato lampeggia in arancione per diversi minuti.
4. Quando l'apparecchio è pronto, la spia di stato lampeggia in verde.

Il ripristino di un dispositivo non lo elimina dal servizio AWS Panorama. Per ulteriori informazioni, consultare Annullamento della registrazione di un'appliance.

Gestione diAWS Panoramaapplicazioni

Le applicazioni vengono eseguite suAWS PanoramaApparecchio per eseguire attività di visione artificiale su flussi video. È possibile creare applicazioni di visione artificiale combinando codice Python e modelli di apprendimento automatico e distribuirli suAWS PanoramaApparecchio su Internet. Le applicazioni possono inviare video a un display o utilizzare l'SDK AWS per inviare risultati ai servizi AWS.

Argomenti

Implementazione di un'applicazione

Per distribuire un'applicazione, devi utilizzare l'interfaccia a riga di comando dell'applicazione AWS Panorama, importarla nel tuo account, creare il contenitore, caricare e registrare risorse e creare un'istanza dell'applicazione. In questo argomento vengono descritti in dettaglio ciascuno di questi passaggi e viene descritto ciò che accade in background.

Se non hai ancora distribuito un'applicazione, consulta Nozioni di base su AWS Panorama la procedura dettagliata.

Per ulteriori informazioni sulla personalizzazione e l'estensione dell'applicazione di esempio, vedere. In fase di creazioneAWS Panoramaapplicazioni

Sezioni

- Installa l'interfaccia a riga di comando dell'applicazione AWS Panorama
- Importazione di un'applicazione
- Crea un'immagine del contenitore
- Importa un modello
- Caricare le risorse dell'applicazione
- Implementa un'applicazione con la console AWS Panorama
- Automatizza la distribuzione delle applicazioni

Installa l'interfaccia a riga di comando dell'applicazione AWS Panorama

Per installare l'interfaccia a riga di comando dell'applicazione AWS PanoramaAWS CLI, usa pip.

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

Per creare immagini di applicazioni con l'interfaccia a riga di comando dell'applicazione AWS Panorama, hai bisogno di Docker. Su Linux sono necessarie anche le librerie di sistema correlate. `qemu` Per ulteriori informazioni sull'installazione e la configurazione dell'interfaccia a riga di comando dell'applicazione AWS Panorama, consulta il file README nel repository del progetto. GitHub

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

Per istruzioni sulla configurazione di un ambiente di compilazione in Windows con WSL2, vedere. Configurazione di un ambiente di sviluppo in Windows

Importazione di un'applicazione

Se lavori con un'applicazione di esempio o un'applicazione fornita da una terza parte, utilizza l'interfaccia a riga di comando dell'applicazione AWS Panorama per importare l'applicazione.

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

Questo comando rinomina i pacchetti delle applicazioni con l'ID del tuo account. I nomi dei pacchetti iniziano con l'ID dell'account su cui sono distribuiti. Quando distribuisci un'applicazione su più account, devi importare e impacchettare l'applicazione separatamente per ogni account.

Ad esempio, l'applicazione di esempio di questa guida, un pacchetto di codice e un pacchetto modello, ciascuno denominato con un ID account segnaposto. Il `import-application` comando li rinomina per utilizzare l'ID account che la CLI deduce dalle credenziali del tuo spazio di lavoro. 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`viene sostituito dall'ID dell'account nei nomi delle directory dei pacchetti e nell'applicazione manifest (`graph.json`), che fa riferimento ad essi. Puoi confermare l'ID del tuo account chiamando `aws sts get-caller-identity` con ilAWS CLI.

```
$ aws sts get-caller-identity
```

```
{
    "UserId": "AIDAXMPL7W66UC3GFXMPL",
    "Account": "210987654321",
    "Arn": "arn:aws:iam::210987654321:user/devenv"
}
```

Crea un'immagine del contenitore

Il codice dell'applicazione è contenuto in un'immagine contenitore Docker, che include il codice dell'applicazione e le librerie che installi nel tuo Dockerfile. Usa il `build-container` comando AWS Panorama Application CLI per creare un'immagine Docker ed esportare un'immagine del file system.

```
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
```

Questo comando crea un'immagine Docker denominata `code_asset` ed esporta un file system in un `.tar.gz` archivio nella cartella. `assets` L'interfaccia a riga di comando estrae l'immagine di base dell'applicazione da Amazon Elastic Container Registry (Amazon ECR), come specificato nel Dockerfile dell'applicazione.

Oltre all'archivio del contenitore, la CLI crea una risorsa per il descrittore del pacchetto ()`descriptor.json`. Entrambi i file vengono rinominati con un identificatore univoco che riflette un hash del file originale. L'interfaccia a riga di comando dell'applicazione AWS Panorama aggiunge anche un blocco alla configurazione del pacchetto che registra i nomi delle due risorse. Questi nomi vengono utilizzati dall'appliance durante il processo di installazione.

Example packages/123456789012-sample_code-1.0/package.json — con blocco di risorse

```
{
    "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"
                    },
```

Il nome della risorsa di codice, specificato nel `build-container` comando, deve corrispondere al valore del `asset` campo nella configurazione del pacchetto. Nell'esempio precedente, entrambi i valori sono`code_asset`.

Importa un modello

L'applicazione potrebbe avere un archivio modello nella cartella delle risorse o che si scarica separatamente. Se hai un nuovo modello, un modello aggiornato o un file descrittore del modello aggiornato, usa il `add-raw-model` comando per importarlo.

```
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
```

Se hai solo bisogno di aggiornare il file descrittore, puoi riutilizzare il modello esistente nella directory assets. Potrebbe essere necessario aggiornare il file descrittore per configurare funzionalità come la modalità di precisione a virgola mobile. Ad esempio, il seguente script mostra come eseguire questa operazione con l'app di esempio.

Example [util-script/ .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
```

Le modifiche al file descrittore nella directory del pacchetto del modello non vengono applicate finché non lo si reimporta con la CLI. La CLI aggiorna la configurazione del pacchetto modello con i nuovi nomi delle risorse, in modo simile a come aggiorna la configurazione del pacchetto di codice dell'applicazione quando si ricostruisce un contenitore.

Caricare le risorse dell'applicazione

Per caricare e registrare le risorse dell'applicazione, che includono l'archivio del modello, l'archivio del file system contenitore e i relativi file descrittori, utilizzate il comando. `package-application`

```
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
...
```

Se non ci sono modifiche a un file di risorse o alla configurazione del pacchetto, la CLI lo ignora.

```
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
```

L'interfaccia a riga di comando carica le risorse per ogni pacchetto su un punto di accesso Amazon S3 specifico per il tuo account. AWS Panorama gestisce il punto di accesso per te e fornisce informazioni su di esso tramite l'DescribePackageAPI. L'interfaccia a riga di comando carica le risorse per ogni pacchetto nella posizione fornita per quel pacchetto e le registra con il servizio AWS Panorama con le impostazioni descritte dalla configurazione del pacchetto.

Implementa un'applicazione con la console AWS Panorama

Puoi distribuire un'applicazione con la console AWS Panorama. Durante il processo di distribuzione, scegli quali stream di videocamere passare al codice dell'applicazione e configuri le opzioni fornite dallo sviluppatore dell'applicazione.

Per distribuire un'applicazione

1. Apri la pagina Applicazioni distribuite sulla console AWS Panorama.
2. Scegli Deploy application.
3. Incolla il contenuto del manifesto dell'applicazione`graph.json`, nell'editor di testo. Seleziona Successivo.
4. Inserisci un nome e una descrizione.
5. Scegli Procedi alla distribuzione.

6. Scegli Inizia la distribuzione.
7. Se la tua applicazione utilizza un ruolo, sceglilo dal menu a discesa. Seleziona Successivo.
8. Scegli Seleziona dispositivo, quindi scegli il tuo elettrodomestico. Seleziona Successivo.
9. Nella fase Seleziona fonti di dati, scegli Visualizza input e aggiungi lo stream della videocamera come fonte di dati. Seleziona Successivo.
10. Nella fase Configurazione, configura tutte le impostazioni specifiche dell'applicazione definite dallo sviluppatore. Seleziona Successivo.
11. Scegli Distribuisci, quindi scegli Fine.
12. Nell'elenco delle applicazioni distribuite, scegli l'applicazione per monitorarne lo stato.

Il processo di installazione richiede 15-20 minuti. L'uscita dell'appliance può rimanere vuota per un periodo prolungato durante l'avvio dell'applicazione. Se riscontri un errore, consultaRisoluzione dei problemi.

Automatizza la distribuzione delle applicazioni

Puoi automatizzare il processo di distribuzione delle applicazioni con l'CreateApplicationInstanceAPI. L'API accetta due file di configurazione come input. Il manifesto dell'applicazione specifica i pacchetti utilizzati e le relative relazioni. Il secondo file è un file di sostituzioni che specifica le sostituzioni dei valori nel manifesto dell'applicazione in fase di distribuzione. L'utilizzo di un file di sostituzione consente di utilizzare lo stesso manifesto dell'applicazione per distribuire l'applicazione con diversi flussi di telecamere e configurare altre impostazioni specifiche dell'applicazione.

Per ulteriori informazioni e script di esempio per ciascuno dei passaggi di questo argomento, vedereAutomatizza la distribuzione delle applicazioni.

Gestione delle applicazioni nella console AWS Panorama

Usa la console AWS Panorama per gestire le applicazioni distribuite.

Sezioni

Aggiornare o copiare un'applicazione

Per aggiornare un'applicazione, utilizzare ilSostituzioneopzione. Quando si sostituisce un'applicazione, è possibile aggiornarne il codice o i modelli.

Per aggiornare un'applicazione

1. Aprire la console AWS PanoramaPagina delle applicazioni distribuite.
2. Scegliere un'applicazione.
3. Scegliere Replace (Sostituisci).
4. Seguire le istruzioni per creare una nuova versione o applicazione.

C'è anche unClonazioneopzione che agisce in modo simile aSostituzione, ma non rimuove la vecchia versione dell'applicazione. È possibile utilizzare questa opzione per testare le modifiche apportate a un'applicazione senza interrompere la versione in esecuzione o per ridistribuire una versione già eliminata.

Eliminare versioni e applicazioni

Per eliminare le versioni delle applicazioni non utilizzate, eliminale dai tuoi dispositivi.

Per eliminare un'applicazione

1. Aprire la console AWS PanoramaPagina delle applicazioni distribuite.
2. Scegliere un'applicazione.
3. ScegliEliminazione dal dispositivo.

Configurazione pacchetti

Quando si utilizza il comando CLI di AWS Panorama Applicationpanorama-cli package-application, la CLI carica le risorse della tua applicazione su Amazon S3 e le registra con AWS Panorama. Le risorse includono file binari (immagini e modelli container) e file descrittori, che AWS Panorama Appliance scarica durante la distribuzione. Per registrare le risorse di un pacchetto, è necessario fornire un file di configurazione del pacchetto separato che definisce il pacchetto, le sue risorse e la relativa interfaccia.

L'esempio seguente mostra una configurazione di pacchetto per un nodo di codice con un input e un output. L'ingresso video consente l'accesso ai dati immagine da un flusso di telecamere. Il nodo di output invia immagini elaborate a un display.

Example Pacchetti/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"
                    }
                ]
            }
        ]
    }
}
```

La`assets`sezione specifica i nomi degli artefatti che l'interfaccia a riga di comando dell'applicazione AWS Panorama ha caricato su Amazon S3. Se importate un'applicazione di esempio o un'applicazione da un altro utente, questa sezione può essere vuota o fare riferimento a risorse che non sono presenti nel vostro account. Quando corri`panorama-cli package-application`, la CLI di AWS Panorama Application popola questa sezione con i valori corretti.

Il manifesto dell'applicazione AWS Panorama

Quando si distribuisce un'applicazione, viene fornito un file di configurazione denominato manifest dell'applicazione. Questo file definisce l'applicazione come un grafico con nodi e bordi. Il manifest dell'applicazione fa parte del codice sorgente dell'applicazione e viene memorizzato nella`graphs`directory.

Example grafi/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"
            }
        ]
    }
}
```

I nodi sono collegati da spigoli, che specificano i mapping tra input e output dei nodi. L'output di un nodo si collega all'input di un altro, formando un grafico.

schema JSON

Il formato dei documenti manifesto e di sostituzione dell'applicazione è definito in uno schema JSON. È possibile utilizzare lo schema JSON per convalidare i documenti di configurazione prima della distribuzione. Lo schema JSON è disponibile in questa guidaGitHubrepository.

- schema JSON–aws-panorama-developer-guida/risorse

Nodi di applicazione

I nodi sono modelli, codice, flussi di telecamere, output e parametri. Un nodo ha un'interfaccia che ne definisce gli ingressi e le uscite. L'interfaccia può essere definita in un pacchetto nel tuo account, in un pacchetto fornito da AWS Panorama o in un tipo incorporato.

Nell'esempio seguente,`code_node`e`model_node`fare riferimento al codice di esempio e ai pacchetti modello inclusi nell'applicazione di esempio.`camera_node`utilizza un pacchetto fornito da AWS Panorama per creare un segnaposto per un flusso di telecamere specificato durante la distribuzione.

Example graph.json — Nodi

```
    "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

I bordi mappano l'output da un nodo a quello di un altro. Nell'esempio seguente, il primo bordo mappa l'output da un nodo di flusso della telecamera all'input di un nodo di codice dell'applicazione. I nomi`video_in`e`video_out`sono definite nelle interfacce dei pacchetti di nodi.

Example graph.json — bordi

```
    "edges": [
```

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

Nel codice di un'applicazione, puoi utilizzare il`inputs`e`outputs`attributi per ottenere le immagini dal flusso di input e inviare le immagini al flusso di output.

Example application.py — Ingresso e uscita video

```
    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)
```

Nodi astratti

In un manifesto dell'applicazione, un nodo astratto fa riferimento a un pacchetto definito da AWS Panorama, che è possibile utilizzare come segnaposto nel manifesto dell'applicazione. AWS Panorama fornisce due tipi di nodo astratto.

- Flusso di telecamere— Scegliere il flusso di telecamere utilizzato dall'applicazione durante la distribuzione.

  Package name (Nome pacchetto)–`panorama::abstract_rtsp_media_source`

  Nome interfaccia–`rtsp_v1_interface`
- Uscita HDMI— Indica che l'applicazione emette video.

  Package name (Nome pacchetto)–`panorama::hdmi_data_sink`

Nome interfaccia–`hdmi0`

Nell'esempio seguente viene illustrato un set di base di pacchetti, nodi e bordi per un'applicazione che elabora flussi di telecamere e trasmette video su un display. Il nodo della fotocamera, che utilizza l'interfaccia del`abstract_rtsp_media_source`pacchetto in AWS Panorama, può accettare più flussi di telecamere come input. Il nodo di output, a cui fa riferimento`hdmi_data_sink`, consente di accedere al codice dell'applicazione a un buffer video che viene emesso dalla porta HDMI dell'appliance.

Example graph.json — Nodi astratti

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

I parametri dell'applicazione

I parametri sono nodi che hanno un tipo base e possono essere ignorati durante la distribuzione. Un parametro può avere un valore predefinito e undecoratore, che indica all'utente dell'applicazione come configurarla.

Tipi di parametro

- `string`: una stringa. Ad esempio, `DEBUG`.
- `int32`: un intero, Ad esempio, `20`
- `float32`: un numero in virgola mobile. Ad esempio, `47.5`
- `boolean–trueofalse`.

L'esempio seguente mostra due parametri, una stringa e un numero, che vengono inviati a un nodo di codice come input.

Example graph.json — Parametri

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

È possibile modificare i parametri direttamente nel manifesto dell'applicazione o fornire nuovi valori al momento della distribuzione con sostituzioni. Per ulteriori informazioni, consultare Configurazione del tempo di distribuzione con sostituzioni.

Configurazione del tempo di distribuzione con sostituzioni

È possibile configurare parametri e nodi astratti durante la distribuzione. Se utilizzi la console AWS Panorama per la distribuzione, puoi specificare un valore per ciascun parametro e scegliere un flusso di telecamere come input. Se utilizzi l'API di AWS Panorama per distribuire le applicazioni, specifichi queste impostazioni con un documento di sostituzione.

Un documento di sostituzione è simile nella struttura a un manifesto dell'applicazione. Per i parametri con tipi di base, si definisce un nodo. Per i flussi di telecamere, è possibile definire un nodo e un pacchetto che vengono mappati a un flusso di telecamere registrato. Quindi si definisce un'override per ogni nodo che specifica il nodo dal manifesto dell'applicazione che sostituisce.

Example overrides.json

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

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

Nell'esempio precedente, il documento definisce le sostituzioni per un parametro stringa e un nodo astratto della fotocamera. La`node0verrides`indica ad AWS Panorama quali nodi di questo documento sovrascrivono quali nel manifesto dell'applicazione.

In fase di creazioneAWS Panoramaapplicazioni

Le applicazioni vengono eseguite suAWS PanoramaApparecchio per eseguire attività di visione artificiale su flussi video. È possibile creare applicazioni di visione artificiale combinando codice Python e modelli di apprendimento automatico e distribuirli nelAWS PanoramaApparecchio su Internet. Le applicazioni possono inviare video a un display o utilizzare l'SDK AWS per inviare risultati ai servizi AWS.

UNmodelloanalizza le immagini per rilevare persone, veicoli e altri oggetti. Sulla base delle immagini che ha visto durante l'allenamento, il modello ti dice cosa pensa che sia qualcosa e quanto è sicuro nella sua ipotesi. Puoi addestrare i modelli con i tuoi dati immagine o iniziare con un campione.

L'applicazionecodiceelabora immagini fisse da un flusso di telecamere, le invia a un modello ed elabora il risultato. Un modello potrebbe rilevare più oggetti e restituirne le forme e la posizione. Il codice può utilizzare queste informazioni per aggiungere testo o grafica al video o per inviare risultati a unAWSservizio per la conservazione o l'ulteriore elaborazione.

Per ottenere immagini da un flusso, interagire con un modello e produrre video, il codice dell'applicazione utilizzaloAWS PanoramaSDK dell'applicazione. L'applicazione SDK è una libreria Python che supporta modelli generati con PyTorch, Apache MXNet e TensorFlow.

Argomenti

- Modelli di visione artificiale
- Creazione di un'immagine dell'applicazione
- Chiamare i servizi AWS dal codice dell'applicazione
- L'SDK dell'applicazione AWS Panorama
- Esecuzione di più thread
- Distribuzione del traffico in entrata
- Utilizzo della GPU
- Configurazione di un ambiente di sviluppo in Windows

Modelli di visione artificiale

Un modello di visione artificiale è un programma software addestrato a rilevare oggetti nelle immagini. Un modello impara a riconoscere un insieme di oggetti analizzando innanzitutto le immagini di tali oggetti attraverso l'addestramento. Un modello di visione artificiale prende un'immagine come input e restituisce informazioni sugli oggetti che rileva, come il tipo di oggetto e la sua posizione. AWS Panorama supporta modelli di visione artificiale creati con PyTorch Apache MXNet e. TensorFlow

> **Note**
>
> Per un elenco di modelli predefiniti che sono stati testati con AWS Panorama, consulta Compatibilità dei modelli.

Sezioni

Utilizzo dei modelli nel codice

Un modello restituisce uno o più risultati, che possono includere le probabilità delle classi rilevate, informazioni sulla posizione e altri dati. L'esempio seguente mostra come eseguire un'inferenza su un'immagine da un flusso video e inviare l'output del modello a una funzione di elaborazione.

Example application.py — Inferenza

```
    def process_media(self, stream):
        """Runs inference on a frame of video."""
        image_data = preprocess(stream.image,self.MODEL_DIM)
        logger.debug('Image data: {}'.format(image_data))
        # Run inference
        inference_start = time.time()
        inference_results = self.call({"data":image_data}, self.MODEL_NODE)
         # Log metrics
        inference_time = (time.time() - inference_start) * 1000
```

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

L'esempio seguente mostra una funzione che elabora i risultati del modello di classificazione di base. Il modello di esempio restituisce una serie di probabilità, che è il primo e unico valore nell'array dei risultati.

Example application.py — Elaborazione dei risultati

```
   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)
```

Il codice dell'applicazione trova i valori con le probabilità più elevate e li associa alle etichette in un file di risorse che viene caricato durante l'inizializzazione.

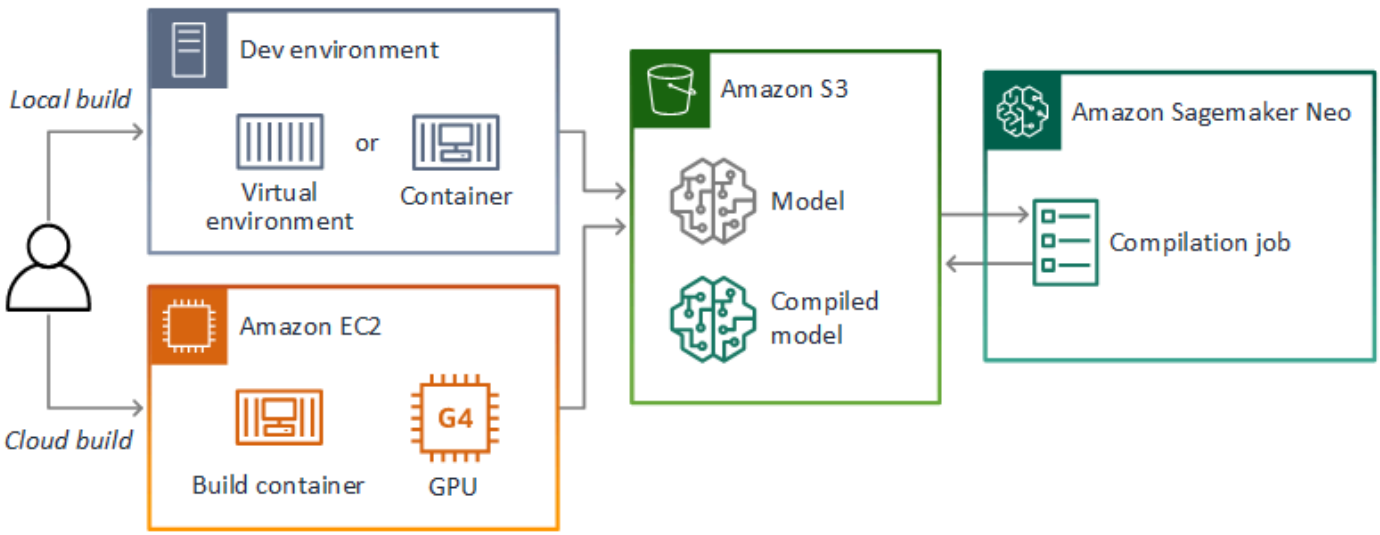
Creazione di un modello personalizzato

Puoi usare modelli che crei in PyTorch Apache MXNet e TensorFlow nelle applicazioni AWS Panorama. In alternativa alla creazione e alla formazione di modelli inSageMaker, puoi utilizzare un modello addestrato o creare e addestrare il tuo modello con un framework supportato ed esportarlo in un ambiente locale o in Amazon EC2.

> **Note**
>
> Per dettagli sulle versioni del framework e sui formati di file supportati da SageMaker Neo, consulta Supported Frameworks nella Amazon SageMaker Developer Guide.

L'archivio di questa guida fornisce un'applicazione di esempio che dimostra questo flusso di lavoro per un modello Keras in formato. TensorFlow `SavedModel` Utilizza TensorFlow 2 e può essere eseguito localmente in un ambiente virtuale o in un contenitore Docker. L'app di esempio include anche modelli e script per la creazione del modello su un'istanza Amazon EC2.

- Esempio di applicazione di modello personalizzato



AWS Panorama utilizza SageMaker Neo per compilare modelli da utilizzare sull'appliance AWS Panorama. Per ogni framework, usa il formato supportato da SageMaker Neo e impacchetta il modello in un `.tar.gz` archivio.

Per ulteriori informazioni, consulta Compilazione e distribuzione di modelli con Neo nella Amazon SageMaker Developer Guide.

Imballaggio di un modello

Un pacchetto modello comprende un descrittore, una configurazione del pacchetto e un archivio del modello. Come in un pacchetto di immagini dell'applicazione, la configurazione del pacchetto indica al servizio AWS Panorama dove sono archiviati il modello e il descrittore in Amazon S3.

Example Pacchetti/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**
>
> Specifica solo la versione principale e secondaria della versione del framework. Per un elenco delle versioni e delle TensorFlow versioni supportate PyTorch di Apache MXNet, vedi Framework supportati.

Per importare un modello, usa il `import-raw-model` comando AWS Panorama Application CLI. Se apportate modifiche al modello o al relativo descrittore, è necessario eseguire nuovamente questo comando per aggiornare le risorse dell'applicazione. Per ulteriori informazioni, consulta Modifica del modello di visione artificiale.

Per lo schema JSON del file descrittore, vedete AssetDescriptor.schema.json.

Modelli di allenamento

Quando addestrate un modello, utilizzate immagini dall'ambiente di destinazione o da un ambiente di test che assomiglia molto all'ambiente di destinazione. Considerate i seguenti fattori che possono influire sulle prestazioni del modello:

- Illuminazione: la quantità di luce riflessa da un soggetto determina la quantità di dettagli che il modello deve analizzare. Un modello addestrato con immagini di soggetti ben illuminati potrebbe non funzionare bene in un ambiente con scarsa illuminazione o retroilluminato.
- Risoluzione: la dimensione di input di un modello è in genere fissata a una risoluzione compresa tra 224 e 512 pixel di larghezza in un rapporto di aspetto quadrato. Prima di passare un fotogramma di video al modello, potete ridimensionarlo o ritagliarlo per adattarlo alle dimensioni richieste.
- Distorsione dell'immagine: la lunghezza focale e la forma dell'obiettivo di una fotocamera possono causare una distorsione delle immagini lontano dal centro dell'inquadratura. La posizione di una fotocamera determina anche quali caratteristiche del soggetto sono visibili. Ad esempio, una fotocamera aerea con obiettivo grandangolare mostrerà la parte superiore di un soggetto quando è al centro dell'inquadratura e una vista inclinata del lato del soggetto man mano che si allontana dal centro.

Per risolvere questi problemi, potete preelaborare le immagini prima di inviarle al modello e addestrare il modello su una più ampia varietà di immagini che riflettono le variazioni negli ambienti reali. Se un modello deve funzionare in situazioni di illuminazione e con una varietà di telecamere, sono necessari più dati per l'allenamento. Oltre a raccogliere più immagini, puoi ottenere più dati di allenamento creando varianti delle immagini esistenti che sono inclinate o con illuminazione diversa.

Creazione di un'immagine dell'applicazione

L'appliance AWS Panorama esegue le applicazioni come file system container esportati da un'immagine creata dall'utente. Specifichi le dipendenze e le risorse della tua applicazione in un Dockerfile che utilizza l'immagine di base dell'applicazione AWS Panorama come punto di partenza.

Per creare un'immagine dell'applicazione, usi Docker e l'interfaccia a riga di comando dell'applicazione AWS Panorama. L'esempio seguente tratto dall'applicazione di esempio di questa guida dimostra questi casi d'uso.

Example Pacchetti/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
```

Vengono utilizzate le seguenti istruzioni Dockerfile.

- `FROM`— Carica l'immagine di base dell'applicazione (`public.ecr.aws/panorama/panorama-application`).
- `WORKDIR`— Imposta la cartella di lavoro sull'immagine. `/panorama`viene utilizzato per il codice dell'applicazione e i file correlati. Questa impostazione persiste solo durante la compilazione e non influisce sulla directory di lavoro dell'applicazione in fase di esecuzione (`/`).
- `COPY`— Copia i file da un percorso locale a un percorso sull'immagine. `COPY . .`copia i file nella directory corrente (la directory del pacchetto) nella directory di lavoro dell'immagine. Ad esempio, il codice dell'applicazione viene copiato da `packages/123456789012-SAMPLE_CODE-1.0/application.py` a. `/panorama/application.py`
- `RUN`— Esegue i comandi della shell sull'immagine durante la compilazione. Una singola `RUN` operazione può eseguire più comandi in sequenza utilizzando `&&` tra i comandi. Questo esempio aggiorna il gestore di `pip` pacchetti e quindi installa le librerie elencate in. `requirements.txt`

Puoi usare altre istruzioni, come `ADD` e`ARG`, che sono utili in fase di compilazione. Le istruzioni che aggiungono informazioni di runtime al contenitore, ad esempio`ENV`, non funzionano con AWS Panorama. AWS Panorama non esegue un contenitore a partire dall'immagine. Utilizza l'immagine solo per esportare un file system, che viene trasferito all'accessorio.

Specifica delle dipendenze

`requirements.txt`è un file dei requisiti di Python che specifica le librerie utilizzate dall'applicazione. L'applicazione di esempio utilizza Open CV e il. AWS SDK for Python (Boto3)

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

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

Il `pip install` comando nel Dockerfile installa queste librerie nella `dist-packages` directory Python sottostante`/usr/local/lib`, in modo che possano essere importate dal codice dell'applicazione.

Archiviazione locale

AWS Panorama riserva la `/opt/aws/panorama/storage` directory per lo storage delle applicazioni. L'applicazione può creare e modificare file in questo percorso. I file creati nella directory di archiviazione persistono anche dopo i riavvii. Le altre posizioni temporanee dei file vengono cancellate all'avvio.

Creazione di risorse di immagini

Quando crei un'immagine per il pacchetto dell'applicazione con l'interfaccia a riga di comando dell'applicazione AWS Panorama, l'interfaccia a riga di comando viene eseguita `docker build` nella directory dei pacchetti. Questo crea un'immagine dell'applicazione che contiene il codice dell'applicazione. La CLI crea quindi un contenitore, ne esporta il file system, lo comprime e lo archivia nella cartella. `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
```

Il blocco JSON nell'output è una definizione di risorsa che l'interfaccia a riga di comando aggiunge alla configurazione del pacchetto (`package.json`) e registra con il servizio AWS Panorama. La CLI copia anche il file descrittore, che specifica il percorso dello script dell'applicazione (il punto di ingresso dell'applicazione).

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

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

Nella cartella assets, il descrittore e l'immagine dell'applicazione sono denominati in base al checksum SHA-256. Questo nome viene utilizzato come identificatore univoco per la risorsa quando viene archiviata in Amazon S3.

Chiamare i servizi AWS dal codice dell'applicazione

Puoi utilizzarliAWS SDK for Python (Boto) per chiamare i servizi AWS dal codice dell'applicazione. Ad esempio, se il tuo modello rileva qualcosa di straordinario, puoi inviare le metriche ad Amazon CloudWatch, inviare una notifica con Amazon SNS, salvare un'immagine su Amazon S3 o richiamare una funzione Lambda per un'ulteriore elaborazione. La maggior parte dei servizi AWS dispone di un'API pubblica che puoi utilizzare con l'SDK AWS.

L'appliance non dispone dell'autorizzazione per accedere ai servizi AWS per impostazione prefirmato. Per concedergli l'autorizzazione, crea un ruolo per l'applicazione e assegnalo all'istanza dell'applicazione durante la distribuzione.

Sezioni

- Uso di Amazon S3
- Utilizzo dell'argomentoAWS IoT MQTT

Uso di Amazon S3

È possibile utilizzare Amazon S3 per archiviare i risultati dell'elaborazione e altri dati dell'applicazione.

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

Utilizzo dell'argomentoAWS IoT MQTT

È possibile inviare messaggi a un SDK for Python (Boto3) per l'invio di un messaggio a un argomento MQTT per l'AWS IoT. Nell'esempio seguente, l'applicazione pubblica un post su un argomento che prende il nome dell'oggetto dell'appliance, che puoi trovare nella AWS IoTconsole.

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

Scegli un nome che indichi l'ID del dispositivo o un altro identificatore di tua scelta. Per pubblicare messaggi, l'applicazione deve essere autorizzata a chiamare`iot:Publish`.

Per monitorare una coda MQTT

1. Apri la pagina Test dellaAWS IoT console.
2. Per Argomento sottoscrizione, immetti il nome dell'argomento. Ad esempio, `panorama/panorama_my-appliance_Thing_a01e373b`.
3. Scegli Subscribe to topic (Effettua sottoscrizione all'argomento).

L'SDK dell'applicazione AWS Panorama

AWS Panorama Application SDK è una libreria Python per lo sviluppo di applicazioni AWS Panorama. Nel tuocodice dell'applicazione, si utilizza AWS Panorama Application SDK per caricare un modello di visione artificiale, eseguire inferenze e inviare video su un monitor.

> **Note**
>
> Per avere la certezza di disporre dell'accesso alle ultime funzionalità dell'SDK dell'AWS Panorama Application,aggiornare il software dell'appliance.

Per informazioni dettagliate sulle classi definite dall'SDK dell'applicazione e sui relativi metodi, vedereDocumriferimento dell'SDK dell'applicazione.

Sezioni

- Aggiunta di testo e caselle per l'output di video

Aggiunta di testo e caselle per l'output di video

Con AWS Panorama SDK, puoi trasmettere uno streaming video su un display. Il video può includere testo e caselle che mostrano l'output del modello, lo stato corrente dell'applicazione o altri dati.

Ciascun oggetto nella`video_in`array è un'immagine proveniente da un flusso di telecamere collegato all'apparecchio. Il tipo di oggetto è`panoramasdk.media`. Dispone di metodi per aggiungere testo e caselle rettangolari all'immagine, che è possibile assegnare all'immagine`video_out`array.

Nell'esempio seguente, l'applicazione di esempio aggiunge un'etichetta per ciascuno dei risultati. Ogni risultato è posizionato nella stessa posizione a sinistra, ma a diverse altezze.

```
        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)
```

Per aggiungere una casella all'immagine di output, utilizzare`add_rect`. Questo metodo prende 4 valori tra 0 e 1, indicando la posizione degli angoli in alto a sinistra e in basso a destra della casella.

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

Esecuzione di più thread

È possibile eseguire la logica dell'applicazione su un thread di elaborazione e utilizzare altri thread per altri processi in background. Ad esempio, puoi creare un threadserve il traffico HTTPper il debug o un thread che monitora i risultati dell'inferenza e invia dati aAWS.

Per eseguire più thread, si utilizza ilModulo di threadingdalla libreria standard Python per creare un thread per ogni processo. L'esempio seguente mostra il ciclo principale dell'applicazione di esempio del server di debug, che crea un oggetto applicazione e lo utilizza per l'esecuzione.trethread.

Example Pacchetti/123456789012-debug_server-1.0/application.pyLoop principale

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

All'uscita di tutti i thread, l'applicazione si riavvia automaticamente. Larun_cvloop elabora le immagini dai flussi di telecamere. Se riceve un segnale di arresto, interrompe il processo di debugger, che esegue un server HTTP e non può arrestarsi. Ogni thread deve gestire i propri errori. Se un errore non viene rilevato e registrato, il thread esce silenziosamente.

Example Pacchetti/123456789012-debug_server-1.0/application.pyLoop di elaborazione

```
    # 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")
```

I thread comunicano tramite l'applicazione`self`oggetto. Per riavviare il ciclo di elaborazione dell'applicazione, il thread del debugger chiama il`stop`Metodo. Questo metodo imposta un`terminate`attributo, che segnala agli altri thread di chiudersi.

Example Pacchetti/123456789012-debug_server-1.0/application.py— Metodo Stop

```
    # Interrupt processing loop
    def stop(self):
        """Signal application to stop processing."""
        logger.info("STOPPING APPLICATION")
        # Signal processes to stop
        self.terminate = True
    # HTTP debug server
    def run_debugger(self):
        """Process debug commands from local network."""
        class ServerHandler(SimpleHTTPRequestHandler):
            # Store reference to application
            application = self
            # Get status
            def do_GET(self):
                """Process GET requests."""
                logger.info('Get request to {}'.format(self.path))
```

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

Distribuzione del traffico in entrata

È possibile monitorare o eseguire il debug delle applicazioni localmente eseguendo un server HTTP insieme al codice dell'applicazione. Per servire il traffico esterno, è possibile mappare le porte di AWS Panorama Appliance alle porte del contenitore dell'applicazione.

> **Important**
>
> Per impostazione predefinita, AWS Panorama Appliance non accetta traffico in entrata su nessuna porta. L'apertura di porte sull'accessorio comporta rischi impliciti per la sicurezza. Quando si utilizza questa funzionalità, è necessario adottare ulteriori misure per proteggere il tuo elettrodomestico dal traffico esterno e comunicazioni sicure tra i clienti autorizzati e l'apparecchio.
> Il codice di esempio incluso in questa guida è a scopo dimostrativo e non implementa l'autenticazione, l'autorizzazione o la crittografia.

È possibile aprire porte nell'intervallo 8000—9000 sull'accessorio. Queste porte, una volta aperte, possono ricevere traffico da qualsiasi client instradabile. Quando si distribuisce l'applicazione, è necessario specificare quali porte aprire e mappare le porte dell'accessorio alle porte del contenitore dell'applicazione. Il software dell'appliance inoltra il traffico al container e invia le risposte al richiedente. Le richieste vengono ricevute sulla porta dell'accessorio specificata e le risposte vengono inviate su una porta effimera casuale.

Configurazione delle porte in entrata

È possibile specificare le mappature delle porte in tre punti nella configurazione dell'applicazione. Il pacchetto di codice `package.json`, si specifica la porta su cui il nodo di codice è in ascolto in un `network` blocco. L'esempio seguente dichiara che il nodo è in ascolto sulla porta 80.

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

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

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

Nel manifesto dell'applicazione, si dichiara una regola di routing che associa una porta dell'accessorio a una porta del contenitore del codice dell'applicazione. L'esempio seguente aggiunge una regola che associa la porta 8080 del dispositivo alla porta 80 del`code_node`container.

Example grafi/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."
            }
        }
    ]
```

Quando si distribuisce l'applicazione, si specificano le stesse regole nella console di AWS Panorama o con un documento di sostituzione passato alCreateApplicationInstanceAPI. È necessario fornire questa configurazione al momento della distribuzione per confermare che si desidera aprire le porte sull'accessorio.

Example grafi/my-app/override.json

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

Se la porta del dispositivo specificata nel manifesto dell'applicazione è utilizzata da un'altra applicazione, è possibile utilizzare il documento di sostituzione per scegliere una porta diversa.

Traffico serv

Con le porte aperte sul contenitore, è possibile aprire un socket o eseguire un server per gestire le richieste in arrivo. La`debug-server`esempio mostra un'implementazione di base di un server HTTP in esecuzione insieme al codice dell'applicazione di visione artificiale.

> **Important**
>
> L'implementazione di esempio non è sicura per l'uso in produzione. Per evitare di rendere il dispositivo vulnerabile agli attacchi, è necessario implementare controlli di sicurezza appropriati nel codice e nella configurazione di rete.

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

```
    # HTTP debug server
```

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

Il server accetta richieste GET al`/status`percorso per recuperare alcune informazioni sull'applicazione. Accetta anche una richiesta POST a`/restart`per riavviare l'applicazione.

Per dimostrare questa funzionalità, l'applicazione di esempio esegue un client HTTP su un thread separato. Il cliente chiama il`/status`percorso sulla rete locale poco dopo l'avvio e riavvia l'applicazione pochi minuti dopo.

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

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

Il ciclo principale gestisce i thread e riavvia l'applicazione all'uscita.

Example Pacchetti/123456789012-debug_server-1.0/application.py— Loop principale

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

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

Per distribuire l'applicazione di esempio, vedere laistruzioni in questa guida GitHub repository.

Utilizzo della GPU

Puoi accedere al processore grafico (GPU) sull'appliance AWS Panorama per utilizzare librerie accelerate da GPU o eseguire modelli di machine learning nel codice dell'applicazione. Per attivare l'accesso alla GPU, aggiungi l'accesso GPU come requisito alla configurazione del pacchetto dopo aver creato il contenitore del codice dell'applicazione.

> **Important**
>
> Se abiliti l'accesso alla GPU, non puoi eseguire i nodi del modello in nessuna applicazione dell'appliance. Per motivi di sicurezza, l'accesso alla GPU è limitato quando l'appliance esegue un modello compilato con Neo. SageMaker Con l'accesso tramite GPU, è necessario eseguire i modelli in nodi di codice applicativo e tutte le applicazioni sul dispositivo condividono l'accesso alla GPU.

Per attivare l'accesso tramite GPU per la tua applicazione, aggiorna la configurazione del pacchetto dopo aver creato il pacchetto con l'interfaccia a riga di comando dell'applicazione AWS Panorama. L'esempio seguente mostra il `requirements` blocco che aggiunge l'accesso GPU al nodo di codice dell'applicazione.

Example package.json con blocco dei requisiti

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

Aggiorna la configurazione del pacchetto tra le fasi di compilazione e di creazione del pacchetto nel flusso di lavoro di sviluppo.

Per implementare un'applicazione con accesso tramite GPU

1. Per creare il contenitore dell'applicazione, usa il `build-container` comando.

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

2. Aggiungi il `requirements` blocco alla configurazione del pacchetto.
3. Per caricare l'asset del contenitore e la configurazione del pacchetto, usa il `package-application` comando.

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

4. Distribuire l'applicazione.

Per applicazioni di esempio che utilizzano l'accesso tramite GPU, visita il aws-panorama-samplesGitHubrepository.

Configurazione di un ambiente di sviluppo in Windows

Per creare un'applicazione AWS Panorama, usi Docker, strumenti da riga di comando e Python. In Windows, è possibile configurare un ambiente di sviluppo utilizzando Docker Desktop con sottosistema Windows per Linux e Ubuntu. Questo tutorial illustra il processo di configurazione per un ambiente di sviluppo che è stato testato con gli strumenti AWS Panorama e le applicazioni di esempio.

Sezioni

Prerequisiti

Per seguire questo tutorial, è necessaria una versione di Windows che supporti Windows Subsystem for Linux 2 (WSL 2).

- Windows 10 versione 1903 e successive (Build 18362 e versioni successive) o Windows 11
- Funzionalità Windows
  - Sottosistema Windows per Linux
  - Hyper-V
  - Piattaforma di macchine virtuali

Questo tutorial è stato sviluppato con le seguenti versioni software.

- Ubuntu 20.04
- Python 3.8.5
- Docker 20.10.8

Installa WSL 2 e Ubuntu

Se si dispone di Windows 10 versione 2004 e successive (Build 19041 e versioni successive), è possibile installare WSL 2 e Ubuntu 20.04 con il seguente comando PowerShell.

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

Per la versione precedente di Windows, segui le istruzioni contenute nella documentazione di WSL 2: Procedura manuale per le versioni precedenti

Installazione di Docker

Per installare Docker Desktop, scaricare ed eseguire il pacchetto di installazione dahub.docker.com. Se riscontri problemi, segui le istruzioni sul sito Web Docker: Backend Docker Desktop WSL 2.

Esegui Docker Desktop e segui il tutorial di prima esecuzione per creare un contenitore di esempio.

> **Note**
>
> Docker Desktop abilita Docker solo nella distribuzione predefinita. Se sono installate altre distribuzioni Linux prima di eseguire questo tutorial, abilitare Docker nella distribuzione Ubuntu appena installata nel menu delle impostazioni di Docker Desktop sottoRisorse,Integrazione WSL.

Configurare Ubuntu

Ora puoi eseguire i comandi Docker nella tua macchina virtuale Ubuntu. Per aprire un terminale della riga di comando, eseguire la distribuzione dal menu di avvio. La prima volta che lo esegui, si configura un nome utente e una password che è possibile utilizzare per eseguire i comandi dell'amministratore.

Per completare la configurazione dell'ambiente di sviluppo, aggiornare il software della macchina virtuale e installare gli strumenti.

Per configurare la macchina virtuale

1. Aggiorna il software fornito con Ubuntu.

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

2. Installa gli strumenti di sviluppo con apt.

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

3. Installa librerie Python con pip.

```
$ pip3 install awscli panoramacli
```

4. Aprire un nuovo terminale e quindi eseguire`aws configure`per configurare ilAWS CLI.

```
$ aws configure
```

   Se non disponi di chiavi di accesso, puoi generarle nellaConsole IAM.

Infine, scarica e importa l'applicazione di esempio.

Per ottenere l'applicazione di esempio

1. Scaricare ed estrarre l'applicazione di esempio.

```
$ 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. Esegui gli script inclusi per testare la compilazione, creare il contenitore dell'applicazione e caricare pacchetti in AWS Panorama.

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

La CLI dell'applicazione AWS Panorama carica i pacchetti e li registra con il servizio AWS Panorama. È possibile oraDistribuzione di un'applicazione di esempiocon la console AWS Panorama.

Fasi successive

Per esplorare e modificare i file di progetto, è possibile utilizzare File Explorer o un ambiente di sviluppo integrato (IDE) che supporta WSL.

Per accedere al file system della macchina virtuale, aprire File explorer e immettere`\\wsl$`nella barra di navigazione. Questa directory contiene un collegamento al file system della macchina virtuale (`Ubuntu-20.04`) e file system per i dati di Docker. Under`Ubuntu-20.04`, la tua directory utente è all'indirizzo`home\`*`username`*.

> **Note**
>
> Per accedere ai file nell'installazione di Windows da Ubuntu, accedi al`/mnt/c`directory. Ad esempio, puoi elencare i file nella directory di download eseguendo`ls /mnt/c/Users/`*`windows-username`*`/Downloads`.

Con Visual Studio Code, è possibile modificare il codice dell'applicazione nell'ambiente di sviluppo ed eseguire comandi con un terminale integrato. Per installare Visual Studio Code, visitacode.visualstudio.com. Dopo l'installazione, aggiungere ilWSL remotoestensione.

Il terminale di Windows è un'alternativa al terminale Ubuntu standard in cui hai eseguito i comandi. Supporta più schede e può eseguire PowerShell, Command Prompt e terminali per qualsiasi altra varietà di Linux installata. Supporta copia e incolla conCtrl+CeCtrl+V, URL cliccabili e altri utili miglioramenti. Per installare Windows Terminal, visitamicrosoft.com.

L'API AWS Panorama

Puoi utilizzare l'API pubblica del servizio AWS Panorama per automatizzare i flussi di lavoro di gestione di dispositivi e applicazioni. Con l'AWS Command Line Interfaceo l'AWSSDK, puoi sviluppare script o applicazioni che gestiscono risorse e distribuzioni. Il GitHub repository di questa guida include script che è possibile utilizzare come punto di partenza per il proprio codice.

- aws-panorama-developer-guide/util-scripts

Sezioni

Automatizza la registrazione del dispositivo

Per effettuare il provisioning di un apparecchio, utilizzare ilProvisionDeviceAPI. La risposta include un file ZIP con la configurazione del dispositivo e le credenziali temporanee. Decodificare il file e salvarlo in un archivio con il prefissocertificates-omni_.

Example provision-device.sh

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

Le credenziali nell'archivio di configurazione scadono dopo 5 minuti. Trasferisci l'archivio sul tuo dispositivo con l'unità USB inclusa.

Per registrare una fotocamera, utilizzare ilCrea nodo dal processo modelloAPI. Questa API accetta una mappa dei parametri del modello per il nome utente, la password e l'URL della fotocamera. È possibile formattare questa mappa come documento JSON utilizzando la manipolazione di stringhe Bash.

Example register-camera.sh

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

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

In alternativa, è possibile caricare la configurazione JSON da un file.

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

Gestisci le appliance con l'API AWS Panorama

Puoi automatizzare le attività di gestione degli appliance con l'API AWS Panorama.

Visualizzazione dei dispositivi

Per ottenere un elenco di dispositivi con ID di dispositivo, utilizza l'ListDevicesAPI.

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

Per ottenere maggiori dettagli su un dispositivo, utilizza l'DescribeDeviceAPI.

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

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

Aggiornamento del software dell'appliance

Se la`LatestSoftware` versione è più recente della`CurrentSoftware`, puoi aggiornare il dispositivo. Usa l'CreateJobForDevicesAPI per creare un processo di aggiornamento over-the-air (OTA).

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

In uno script, puoi compilare il campo della versione dell'immagine nel file di configurazione del lavoro con la manipolazione delle stringhe Bash.

Example check-updates.sh

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

L'appliance scarica la versione del software specificata e si aggiorna automaticamente. Guarda i progressi dell'aggiornamento con l'DescribeDeviceJobAPI.

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

Per ottenere un elenco di tutti i processi in esecuzione, utilizza il ListDevicesJobs.

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

Per uno script di esempio che verifica e applica gli aggiornamenti, vedi check-updates.sh nel GitHub repository di questa guida.

Riavvio delle appliance

Per riavviare un dispositivo, usa l'CreateJobForDevicesAPI.

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

```
    ]
}
```

In uno script, puoi ottenere un elenco di dispositivi e sceglierne uno da riavviare in modo interattivo.

Example reboot-device.sh — utilizzo

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

Automatizza la distribuzione delle applicazioni

Per distribuire un'applicazione, utilizzi sia l'interfaccia a riga di comando dell'applicazione AWS Panorama cheAWS Command Line Interface. Dopo aver creato il contenitore dell'applicazione, lo carichi insieme ad altre risorse su un punto di accesso Amazon S3. Quindi si distribuisce l'applicazione conCreateApplicationInstanceAPI.

Per ulteriori informazioni e istruzioni sull'uso degli script mostrati, segui le istruzioni contenute nelapplicazione di esempio README.

Sezioni

- Costruisci il contenitore
- Carica il contenitore e registra i nodi
- Distribuzione dell'applicazione
- Monitora la distribuzione

Costruisci il contenitore

Per creare il contenitore dell'applicazione, utilizzarebuild-containercomando. Questo comando crea un contenitore Docker e lo salva come file system compresso nelassetscartella.

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
```

È inoltre possibile utilizzare il completamento della riga di comando per inserire l'argomento del percorso digitando una parte del percorso e quindi premendoTAB.

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

Carica il contenitore e registra i nodi

Per caricare l'applicazione, usapackage-applicationcomando. Questo comando carica risorse daassetscartella in un punto di accesso Amazon S3 gestito da AWS Panorama.

Example 4-package-app.sh

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

L'interfaccia a riga di comando dell'applicazione AWS Panorama carica le risorse del contenitore e del descrittore a cui fa riferimento la configurazione del pacchetto (`package.json`) in ogni pacchetto e registra i pacchetti come nodi in AWS Panorama. Quindi fai riferimento a questi nodi nel manifesto dell'applicazione (`graph.json`) per distribuire l'applicazione.

Distribuzione dell'applicazione

Per distribuire l'applicazione, è necessario utilizzareCreateApplicationInstanceAPI. Questa azione richiede, tra gli altri, i seguenti parametri.

- `ManifestPayload`— Il manifesto dell'applicazione (`graph.json`) che definisce i nodi, i pacchetti, i bordi e i parametri dell'applicazione.
- `ManifestOverridesPayload`— Un secondo manifest che sovrascrive i parametri del primo. Il manifesto dell'applicazione può essere considerato come una risorsa statica nell'origine dell'applicazione, in cui il manifesto di sostituzione fornisce impostazioni in fase di distribuzione che personalizzano la distribuzione.
- `DefaultRuntimeContextDevice`— Il dispositivo bersaglio.
- `RuntimeRoleArn`— L'ARN di un ruolo IAM utilizzato dall'applicazione per accedere ai servizi e alle risorse AWS.
- `ApplicationInstanceIdToReplace`— L'ID di un'istanza di applicazione esistente da rimuovere dal dispositivo.

I payload manifest e override sono documenti JSON che devono essere forniti come valore di stringa annidato all'interno di un altro documento. Per fare ciò, lo script carica i manifesti da un file come stringa e utilizza ilstrumento jqper costruire il documento annidato.

Example 5-deploy.sh— comporre manifesti

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

Lo script di distribuzione utilizzaListDevicesAPI per ottenere un elenco di dispositivi registrati nella regione corrente e salva la scelta dell'utente in un file locale per le distribuzioni successive.

Example 5-deploy.sh— trova un dispositivo

```
    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)
```

Il ruolo dell'applicazione viene creato da un altro script (1-create-role.sh). Lo script di distribuzione ottiene l'ARN di questo ruolo daAWS CloudFormation. Se l'applicazione è già distribuita sul dispositivo, lo script ottiene l'ID dell'istanza dell'applicazione da un file locale.

Example 5-deploy.sh— ruolo, ARN e argomenti sostitutivi

```
# 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
```

Infine, lo script mette insieme tutti i pezzi per creare un'istanza dell'applicazione e distribuire l'applicazione sul dispositivo. Il servizio risponde con un ID di istanza che lo script memorizza per un uso successivo.

Example 5-deploy.sh— distribuire l'applicazione

```
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
```

Monitora la distribuzione

Per monitorare una distribuzione, utilizzareListApplicationInstancesAPI. Lo script di monitoraggio ottiene l'ID del dispositivo e l'ID dell'istanza dell'applicazione dai file nella directory dell'applicazione e li utilizza per creare un comando CLI. Quindi chiama in loop.

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
```

Al termine della distribuzione, puoi visualizzare i registri chiamando AmazonCloudWatchAPI dei log. Lo script view logs utilizza ilCloudWatchRegistri`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
```

Gestisci le applicazioni con l'API AWS Panorama

È possibile monitorare e gestire le applicazioni con l'API AWS Panorama.

Visualizzazione dell'applicazione

Per ottenere un elenco delle applicazioni in esecuzione su un dispositivo, utilizza l'ListApplicationInstancesAPI.

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

Per ottenere maggiori dettagli sui nodi di un'istanza di applicazione, utilizza l'ListApplicationInstanceNodeInstancesAPI.

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

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

Gestisci i flussi delle telecamere

Puoi mettere in pausa e riprendere i nodi dello streaming della fotocamera con l'SignalApplicationInstanceNodeInstancesAPI.

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

In uno script, puoi ottenere un elenco di nodi e sceglierne uno da mettere in pausa o riprendere in modo interattivo.

Example pause-camera.sh — utilizzo

```
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"
}
```

Sospendendo e riprendendo i nodi della telecamera, è possibile passare attraverso un numero di stream di telecamere maggiore di quelli che possono essere elaborati simultaneamente. Per fare ciò, mappa più flussi di telecamere sullo stesso nodo di input nel tuo manifest di override.

Nell'esempio seguente, il manifest di override mappa due stream di telecamere`warehouse-floor` e`entrance-north` allo stesso nodo di input (`camera_node`). Lo`warehouse-floor` stream è attivo all'avvio dell'applicazione e il`entrance-north` nodo attende l'attivazione di un segnale.

Example sovrascrivere 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"
                    }
                ]
            }
```

Per i dettagli sulla distribuzione con l'API, consultaAutomatizza la distribuzione delle applicazioni.

Utilizzo di endpoint VPC

Se lavori in un VPC senza accesso a Internet, puoi creare un endpoint VPC da utilizzare con AWS Panorama. Un endpoint VPC consente ai client in esecuzione in una sottorete privata di connettersi a un servizio AWS senza una connessione Internet.

Per dettagli sulle porte e gli endpoint utilizzati da AWS Panorama Appliance, consulta. ???

Sections

Creazione di un endpoint VPC

Per stabilire una connessione privata tra il tuo VPC e AWS Panorama, crea un endpoint VPC. Non è necessario un endpoint VPC per utilizzare AWS Panorama. Devi creare un endpoint VPC solo se lavori in un VPC senza accesso a Internet. Quando la CLI o l'SDK di AWS tenta di connettersi ad AWS Panorama, il traffico viene instradato attraverso l'endpoint VPC.

Crea un endpoint VPC per AWS Panorama utilizzando le seguenti impostazioni:

- Nome del servizio: **`com.amazonaws.`*`us-west-2`*`.panorama`**
- Tipo: interfaccia

Un endpoint VPC utilizza il nome DNS del servizio per ottenere traffico dai client SDK AWS senza alcuna configurazione aggiuntiva. Per ulteriori informazioni sull'uso degli endpoint VPC, consulta Interface VPC endpoint nella Amazon VPC User Guide.

Connessione di un'appliance a una sottorete privata

L'AWS Panorama Appliance può connettersi AWS tramite una connessione VPN privata con AWS Site-to-Site VPN oAWS Direct Connect. Con questi servizi, puoi creare una sottorete privata che si estende fino al tuo data center. L'appliance si connette alla sottorete privata e accede ai servizi AWS tramite endpoint VPC.

VPN da sito a sito sono servizi per connettere il tuo data center ad Amazon VPC in modo sicuro. AWS Direct Connect Con la VPN Site-to-Site, puoi utilizzare dispositivi di rete disponibili in commercio per connetterti. AWS Direct Connectutilizza un dispositivo per connettersiAWS.

- VPN da sito a sito: cos'è? AWS Site-to-Site VPN
- AWS Direct Connect— Che cos'è? AWS Direct Connect

Dopo aver collegato la rete locale a una sottorete privata in un VPC, crea endpoint VPC per i seguenti servizi.

- Amazon Simple Storage Service, AWS PrivateLinkper Amazon S3
- AWS IoT Core— Utilizzo AWS IoT Core con endpoint VPC di interfaccia (piano dati e fornitore di credenziali)
- Amazon Elastic Container Registry — Endpoint VPC dell'interfaccia Amazon Elastic Container Registry
- Amazon CloudWatch: utilizzo CloudWatch con endpoint VPC di interfaccia
- Amazon CloudWatch Logs: utilizzo dei CloudWatch log con endpoint VPC di interfaccia

L'appliance non necessita di connettività al servizio AWS Panorama. Comunica con AWS Panorama tramite un canale di messaggistica inAWS IoT.

Oltre agli endpoint VPC, Amazon S3 richiede AWS IoT l'uso di zone private ospitate su Amazon Route 53. La zona ospitata privata indirizza il traffico dai sottodomini, inclusi i sottodomini per i punti di accesso Amazon S3 e gli argomenti MQTT, all'endpoint VPC corretto. Per informazioni sulle zone ospitate private, consulta Working with private hosted zones nella Amazon Route 53 Developer Guide.

Per una configurazione VPC di esempio con endpoint VPC e zone ospitate private, consulta. Modelli di esempio AWS CloudFormation

Modelli di esempio AWS CloudFormation

L' GitHub archivio di questa guida fornisce AWS CloudFormation modelli che puoi utilizzare per creare risorse da utilizzare con AWS Panorama. I modelli creano un VPC con due sottoreti private, una sottorete pubblica e un endpoint VPC. È possibile utilizzare le sottoreti private nel VPC per

ospitare risorse isolate da Internet. Le risorse nella sottorete pubblica possono comunicare con le risorse private, ma non è possibile accedervi da Internet.

Example vpc-endpoint.yml — Sottoreti private

```
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
  ...
```

Il `vpc-endpoint.yml` modello mostra come creare un endpoint VPC per AWS Panorama. Puoi utilizzare questo endpoint per gestire le risorse AWS Panorama con l'AWSSDK o. AWS CLI

Example vpc-endpoint.yml — endpoint VPC

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

```
      - !GetAtt vpc.DefaultSecurityGroup
      PrivateDnsEnabled: true
      SubnetIds:
      - !Ref privateSubnetA
      - !Ref privateSubnetB
      PolicyDocument:
        Version: 2012-10-17
        Statement:
        - Effect: Allow
          Principal: "*"
          Action:
            - "panorama:*"
          Resource:
            - "*"
```

`PolicyDocument`È una politica di autorizzazioni basata sulle risorse che definisce le chiamate API che possono essere effettuate con l'endpoint. È possibile modificare la policy per limitare le azioni e le risorse a cui è possibile accedere tramite l'endpoint. Per ulteriori informazioni, consulta Controllo degli accessi ai servizi con endpoint VPC nella Guida per l'utente di Amazon VPC.

Il `vpc-appliance.yml` modello mostra come creare endpoint VPC e zone private ospitate per i servizi utilizzati da AWS Panorama Appliance.

Example vpc-appliance.yml — Endpoint del punto di accesso Amazon S3 con zona ospitata privata

```
  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 ] ] ]
```

I modelli di esempio illustrano la creazione di risorse Amazon VPC e Route 53 con un VPC di esempio. Puoi adattarli al tuo caso d'uso rimuovendo le risorse VPC e sostituendo i riferimenti a sottorete, gruppo di sicurezza e ID VPC con gli ID delle tue risorse.

Applicazioni, script e modelli di esempio

La GitHub L'archivio di questa guida fornisce applicazioni, script e modelli di esempio perAWS Panoramadispositivi. Usa questi esempi per apprendere le migliori pratiche e automatizzare i flussi di lavoro di sviluppo.

Sezioni

- Applicazioni di esempio
- script di utilità
- AWS CloudFormationModelli di
- Altri esempi e strumenti

Applicazioni di esempio

Le applicazioni di esempio dimostrano l'uso diAWS Panoramacaratteristiche e attività comuni di visione artificiale. Queste applicazioni di esempio includono script e modelli che automatizzano la configurazione e la distribuzione. Con una configurazione minima, è possibile distribuire e aggiornare le applicazioni dalla riga di comando.

- aws-panorama-sample— Visione artificiale di base con un modello di classificazione. Usa ilAWS SDK for Python (Boto)per caricare le metriche su CloudWatch, metodi di preelaborazione e inferenza degli strumenti e configurazione della registrazione.
- server di debug—Porte in ingresso aperteSul dispositivo e inoltra il traffico a un contenitore di codice applicativo. Utilizza il multithreading per eseguire contemporaneamente il codice dell'applicazione, un server HTTP e un client HTTP.
- modello personalizzato— Esporta modelli dal codice e compila con SageMaker Neo per testare la compatibilità conAWS PanoramaApparecchio. Crea localmente in uno sviluppo Python, in un contenitore Docker o su un'istanza Amazon EC2. Esporta e compila tutti i modelli applicativi integrati in Keras per uno specifico TensorFlow o versione Python.

Per altre applicazioni di esempio, visita anche ilaws-panorama-samplesarchivio.

script di utilità

Gli script in`util-scripts`gestione delle cartelleAWS Panoramarisorse o automatizza i flussi di lavoro di sviluppo.

- provision-device.sh— Esegui il provisioning di un dispositivo.
- check-updates.sh— Verifica e applica gli aggiornamenti software dell'appliance.
- reboot-device.sh— Riavviare un dispositivo.
- register-camera.sh— Registra una videocamera.
- deregister-camera.sh— Eliminare un nodo videocamera.
- view-logs.sh— Visualizza i log per un'istanza dell'applicazione.
- pause-camera.sh— Mette in pausa o riprende lo streaming di una videocamera.
- push.sh— Crea, carica e distribuisci un'applicazione.
- rename-package.sh— Rinomina un pacchetto di nodi. Aggiorna i nomi delle directory, i file di configurazione e il manifesto dell'applicazione.
- samplify.sh— Sostituisci l'ID dell'account con un ID account di esempio e ripristina le configurazioni di backup per rimuovere la configurazione locale.
- update-model-config.sh— Aggiungete nuovamente il modello all'applicazione dopo aver aggiornato il file descrittore.
- cleanup-patches.sh— Annulla la registrazione delle vecchie versioni delle patch ed elimina i relativi manifesti da Amazon S3.

Per i dettagli sull'utilizzo, consultail file README.

AWS CloudFormationModelli di

Usa ilAWS CloudFormationmodelli in`cloudformation-templates`directory per cui creare risorseAWS Panoramaapplicazioni.

- alarm-application.yml— Crea un allarme che monitora gli errori di un'applicazione. Se l'istanza dell'applicazione genera errori o smette di funzionare per 5 minuti, l'allarme invia un'e-mail di notifica.

- alarm-device.yml— Crea un allarme che monitora la connettività di un dispositivo. Se il dispositivo smette di inviare metriche per 5 minuti, l'allarme invia un'e-mail di notifica.
- ruolo-applicazione.yml— Creare un ruolo applicativo. Il ruolo include l'autorizzazione a inviare metriche a CloudWatch. Aggiungi le autorizzazioni all'informativa per altre operazioni API utilizzate dall'applicazione.
- vpc-appliance.yml— Creare un VPC con accesso privato al servizio di sottorete perAWS PanoramaApparecchio. Per collegare l'appliance a un VPC, utilizzareAWS Direct ConnectoAWS Site-to-Site VPN.
- vpc-endpoint.yml— Creare un VPC con accesso privato al servizio di sottoreteAWS Panoramaservizio. Le risorse all'interno del VPC possono connettersi aAWS Panoramaa monitorare e gestireAWS Panoramarisorse senza connessione a Internet.

La`create-stack.sh`lo script in questa directory creaAWS CloudFormationpile. Richiede un numero variabile di argomenti. Il primo argomento è il nome del modello e gli argomenti rimanenti sostituiscono i parametri del modello.

Ad esempio, il comando seguente crea un ruolo dell'applicazione.

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

Altri esempi e strumenti

Ilaws-panorama-samplesil repository contiene più applicazioni di esempio e strumenti utili.

- Applicazioni— Applicazioni di esempio per varie architetture di modelli e casi d'uso.
- Validazione del flusso della telecamera— Convalida i flussi della videocamera.
- PanoJupyter— Esegui JupyterLab su unAWS Panoramaapparecchio.
- Caricamento laterale— Aggiorna il codice dell'applicazione senza creare o distribuire un contenitore di applicazioni.

- IlAWSla comunità ha anche sviluppato strumenti e linee guida perAWS Panorama. Dai un'occhiata ai seguenti progetti open source su GitHub.

- tagliabiscotti - panorama— Un modello Cookiecutter perAWS Panoramaapplicazioni.

- zaino— Moduli Python per accedere ai dettagli dell'ambiente di runtime, alla profilazione e alle opzioni di uscita video aggiuntive.

MonitoraggioAWS Panoramarisorse e applicazioni

È possibile monitorareAWS Panoramarisorse nelAWS Panoramaconsole e con AmazonCloudWatch. LaAWS PanoramaL'apparecchio si collega alAWSCloud su Internet per segnalare lo stato e lo stato delle telecamere collegate. Mentre è acceso, l'apparecchio invia anche i registri aCloudWatchEffettua il login in tempo reale.

L'apparecchio ottiene l'autorizzazione per l'usoAWS IoT,CloudWatchRegistri e altri servizi AWS da un ruolo di servizio che crei la prima volta che utilizziAWS Panoramaconsole. Per ulteriori informazioni, consultare . Ruoli di servizio AWS Panorama e risorse interservizi .

Per assistenza nella risoluzione di errori specifici, consultaRisoluzione dei problemi.

Argomenti

- Monitoraggio nella console AWS Panorama
- Visualizzazione di log AWS Panorama
- Monitoraggio di dispositivi e applicazioni con AmazonCloudWatch

Monitoraggio nella console AWS Panorama

È possibile utilizzare la console AWS Panorama per monitorare AWS Panorama Appliance e le telecamere. La console utilizzaAWS IoTper monitorare lo stato dell'apparecchio.

Per monitorare l'appliance nella console AWS Panorama

1. Apertura dellaConsole AWS Panorama.
2. Aprire la console AWS PanoramaLa pagina dispositivi.
3. Scegli un apparecchio.
4. Per visualizzare lo stato di un'istanza di applicazione, selezionala dall'elenco.
5. Per visualizzare lo stato delle interfacce di rete dell'appliance, scegliereImpostazioni.

Lo stato generale dell'apparecchio viene visualizzato nella parte superiore della pagina. Se lo stato di èOnline, quindi l'apparecchio è collegato aAWSe l'invio di aggiornamenti di stato regolari.

Visualizzazione di log AWS Panorama

AWS Panorama segnala gli eventi delle applicazioni e del sistema ad Amazon CloudWatch Registri. In caso di problemi, puoi utilizzare i registri degli eventi per eseguire il debug dell'applicazione AWS Panorama o per risolvere i problemi di configurazione dell'applicazione.

Per visualizzare i log CloudWatch Log

1. Aprire ilPagina dei gruppi di log della CloudWatch Console di log.
2. Trova i log delle applicazioni e delle appliance AWS Panorama nei seguenti gruppi:

   - Registri del dispositivo–`/aws/panorama/devices/`*`device-id`*
   - Registri delle applicazioni–`/aws/panorama/devices/`*`device-id`*`/applications/`*`instance-id`*

Quando si effettua il riprovisioning di un dispositivo dopo aver aggiornato il software di sistema, è possibile:visualizzare i log nell'unità USB di provisioning.

Sezioni

Visualizzazione di registri del dispositivo

L'appliance AWS Panorama crea un gruppo di log per il dispositivo e un gruppo per ogni istanza dell'applicazione che distribuisci. I registri dei dispositivi contengono informazioni sullo stato dell'applicazione, sugli aggiornamenti software e sulla configurazione del sistema.

Registri del dispositivo:**`/aws/panorama/devices/`*`device-id`***

- `occ_log`— Uscita dal processo del controller. Questo processo coordina le distribuzioni delle applicazioni e riporta lo stato dei nodi di ogni istanza dell'applicazione.
- `ota_log`— Uscita dal processo che coordina over-the-air aggiornamenti software (OTA).

- `syslog`— Uscita dal processo syslog del dispositivo, che acquisisce i messaggi inviati tra i processi.
- `kern_log`— Eventi dal kernel Linux del dispositivo.
- `logging_setup_logs`— Uscita dal processo che configura il CloudWatch Registra l'agente.
- `cloudwatch_agent_logs`— Uscita dal CloudWatch Registra l'agente.
- `shadow_log`— Uscita dalAWS IoTDevice Device.

Visualizzazione di registri dell'applicazione

Il gruppo di log di un'istanza dell'applicazione contiene un flusso di log per ogni nodo, con il nome del nodo.

Registri delle applicazioni —**`/aws/panorama/devices/`*`device-id`*`/applications/`*`instance-id`***

- Codice— Output dal codice dell'applicazione e dall'SDK dell'applicazione AWS Panorama. Aggrega i log delle applicazioni da`/opt/aws/panorama/logs`.
- Modello— Uscita del processo che coordina le richieste di inferenza con un modello.
- Stream— Uscita dal processo che decodifica il video proveniente dallo streaming di una telecamera.
- Visualizzazione— Uscita dal processo che esegue il rendering dell'uscita video per la porta HDMI.
- `mds`— Registri dal server di metadati dell'appliance.
- `console_output`— Acquisisce flussi di output e errori standard dai contenitori di codice.

Se non vengono visualizzati i dati di accesso CloudWatch Registri: verificare di trovarsi nella regione AWS corretta. In tal caso, potrebbe esserci un problema con la connessione dell'appliance ad AWS o con le autorizzazioni sudell'elettrodomesticoAWS Identity and Access ManagementRuolo (IAM).

Configurazione dei registri dell'applicazione

Configura un logger Python per scrivere file di log`/opt/aws/panorama/logs`. L'appliance trasmette i log da questa posizione a CloudWatch Registri. Per evitare di utilizzare troppo spazio su disco, utilizzate una dimensione massima del file di 10 MiB e un numero di backup di 1. Il seguente esempio mostra un metodo che crea un logger.

Example application.py— Configurazione del logger

```
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
```

Inizializza il logger nell'ambito globale e utilizzalo in tutto il codice dell'applicazione.

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()
```

Visualizzazione di registri di provisioning

Durante il provisioning, AWS Panorama Appliance copia i log sull'unità USB utilizzata per trasferire l'archivio di configurazione sull'appliance. Utilizza questi registri per risolvere i problemi di provisioning sui dispositivi con la versione software più recente.

> **Important**
>
> I log di provisioning sono disponibili per i dispositivi aggiornati alla versione software 4.3.23 o successiva.

Log di applicazioni

- `/panorama/occ.log`— Registri del software del controller AWS Panorama.
- `/panorama/ota_agent.log`— AWS Panorama over-the-air aggiorna i registri degli agenti.
- `/panorama/syslog.log`— Log di sistema Linux.
- `/panorama/kern.log`— Registri del kernel Linux.

Estrazione dei log da un dispositivo

Se i registri del dispositivo e delle applicazioni non vengono visualizzati in CloudWatch Registri, è possibile utilizzare un'unità USB per ottenere un'immagine di registro crittografata dal dispositivo. Il team di assistenza AWS Panorama può decrittografare i log per tuo conto e aiutarti nel debug.

Prerequisiti

Per seguire la procedura è necessario il seguente hardware:

- unità USB— Un'unità di memoria flash USB in formato FAT32 con almeno 1 GB di spazio di archiviazione, per trasferire i file di registro dall'AWS Panorama Appliance.

Per estrarre i registri dal dispositivo

1. Preparare un'unità USB con`managed_logs`cartella all'interno di una`panorama`cartella.

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

2. Connect l'unità USB al dispositivo.
3. Spegnimentol'appliance AWS Panorama.
4. Accendi l'appliance AWS Panorama.

5. Il dispositivo copia i registri sul dispositivo. Il LED di statolampeggia in blumentre è in corso.
6. I file di registro possono quindi essere trovati all'internomanaged_logscartella con il formatopanorama_device_log_*v1_dd_hh_mm*.img

Non puoi decifrare tu stesso l'immagine del registro. Collabora con l'assistenza clienti, un account manager tecnico per AWS Panorama o un architetto di soluzioni per coordinarti con il team di assistenza.

Monitoraggio di dispositivi e applicazioni con AmazonCloudWatch

Quando un'appliance è online, AWS Panorama invia le metriche ad AmazonCloudWatch. È possibile creare grafici e dashboard con queste metriche nelCloudWatchconsole per monitorare l'attività dell'appliance e impostare allarmi che notificano all'utente quando i dispositivi sono offline o le applicazioni riscontrano errori.

Per visualizzare i parametri nella console CloudWatch

1. Apertura dellaPagina Metriche della console AWS Panorama(`PanoramaDeviceMetrics`spazio dei nomi).
2. Scegliere uno schema di quota.
3. Scegliere i parametri per aggiungerli al grafico.
4. Per scegliere un parametro diverso e personalizzare il grafico, utilizzare le opzioni nella scheda Graphed metrics (Parametri grafico). Per impostazione predefinita, i grafici utilizzano la statistica `Average` per tutti i parametri.

> **Prezzi**
>
> CloudWatchha un livello Always Free. Oltre la soglia del piano gratuito, CloudWatch prevede dei costi per parametri, pannelli, allarmi, registri e informazioni dettagliate. Per informazioni dettagliate, consulta Prezzi di CloudWatch.

Per ulteriori informazioni suCloudWatch, consultaAmazonCloudWatchGuida per l'utente di.

Sezioni

- Utilizzo dei parametri dei dispositivi
- Utilizzo delle metriche applicative
- Configurazione degli allarmi

Utilizzo dei parametri dei dispositivi

Quando un dispositivo è online, i parametri vengono forniti ad AmazonCloudWatch. È possibile utilizzare queste metriche per monitorare l'attività del dispositivo e attivare un allarme se i dispositivi sono offline.

- `DeviceActive`— Inviato periodicamente quando il dispositivo è attivo.

  Dimensioni —`DeviceIdeDeviceName`.

Visualizzare il`DeviceActive`parametro con`Average`statistico.

Utilizzo delle metriche applicative

Quando un'applicazione rileva un errore, i parametri vengono visualizzati ad AmazonCloudWatch. Queste parametri consentono di attivare un allarme se un'applicazione interrompe l'esecuzione.

- `ApplicationErrors`— Il numero di errori dell'applicazione registrati.

  Dimensioni —`ApplicationInstanceNameeApplicationInstanceId`.

Visualizzazione dei parametri dell'applicazione con`Sum`statistico.

Configurazione degli allarmi

Per ricevere notifiche quando una metrica supera una soglia, crea un allarme. Ad esempio, puoi creare un allarme che invia una notifica quando la somma del`ApplicationErrors`la metrica rimane a 1 per 20 minuti.

Per creare un allarme

1. Apertura dellaAmazonCloudWatchPagina degli allarmi della console.
2. Scegli Create Alarm (Crea allarme).
3. ScegliereSelezionare parametroe individua una metrica per il tuo dispositivo, ad esempio`ApplicationErrors`per`applicationInstance-gk75xmplqbqtenlnmz4ehiu7xa`,`my-application`.
4. Seguire le istruzioni per configurare una condizione, un'azione e un nome per l'allarme.

Per istruzioni dettagliate, consultaCreazione di unCloudWatchallarmenellaAmazonCloudWatchGuida per l'utente di.

Risoluzione dei problemi

I seguenti argomenti forniscono consigli per la risoluzione di errori e problemi che potrebbero verificarsi durante l'utilizzo della AWS Panorama console, dell'appliance o dell'SDK. Se trovi un problema che non è elencato qui, utilizza il pulsante Fornisci feedback in questa pagina per segnalarlo.

Puoi trovare i log del tuo dispositivo nella console Amazon CloudWatch Logs. L'appliance carica i log dal codice dell'applicazione, dal software dell'appliance e dai processi man mano che vengono generati. AWS IoT Per ulteriori informazioni, consulta Visualizzazione di log AWS Panorama.

Fornitura

Problema: (macOS) Il mio computer non riconosce l'unità USB inclusa con un adattatore USB-C.

Ciò può verificarsi se colleghi l'unità USB a un adattatore USB-C già collegato al computer. Prova a scollegare l'adattatore e a ricollegarlo con l'unità USB già collegata.

Problema: il provisioning non riesce quando utilizzo la mia unità USB.

Problema: il provisioning non riesce quando si utilizza la porta USB 2.0 dell'appliance.

L'AWS Panoramaappliance è compatibile con dispositivi di memoria flash USB di dimensioni comprese tra 1 e 32 GB, ma non tutti sono compatibili. Sono stati riscontrati alcuni problemi durante l'utilizzo della porta USB 2.0 per il provisioning. Per risultati coerenti, utilizzate l'unità USB inclusa con la porta USB 3.0 (accanto alla porta HDMI).

Per il Lenovo ThinkEdge® SE70, l'unità USB non è inclusa nell'appliance. Utilizzare un'unità USB 3.0 con almeno 1 GB di spazio di archiviazione.

Configurazione dell'appliance

Problema: l'appliance mostra una schermata vuota durante l'avvio.

Dopo aver completato la sequenza di avvio iniziale, che richiede circa un minuto, l'appliance mostra una schermata vuota per un minuto o più mentre carica il modello e avvia l'applicazione. Inoltre, l'appliance non emette video se si collega uno schermo dopo l'accensione.

Problema: l'apparecchio non risponde quando tengo premuto il pulsante di accensione per spegnerlo.

L'apparecchio impiega fino a 10 secondi per spegnersi in sicurezza. È necessario tenere premuto il pulsante di accensione solo per 1 secondo per avviare la sequenza di spegnimento. Per un elenco completo delle operazioni dei pulsanti, vedere. Pulsanti e luci di AWS Panorama

Problema: devo generare un nuovo archivio di configurazione per modificare le impostazioni o sostituire un certificato smarrito.

AWS Panoramanon memorizza il certificato del dispositivo o la configurazione di rete dopo averlo scaricato e non è possibile riutilizzare gli archivi di configurazione. Elimina l'appliance utilizzando la AWS Panorama console e creane una nuova con un nuovo archivio di configurazione.

Configurazione dell'applicazione

Problema: quando eseguo più applicazioni, non riesco a controllare quale utilizza l'uscita HDMI.

Quando si distribuiscono più applicazioni con nodi di output, l'applicazione avviata più di recente utilizza l'uscita HDMI. Se l'applicazione smette di funzionare, l'output può essere utilizzato da un'altra applicazione. Per consentire a una sola applicazione di accedere all'output, rimuovete il nodo di output e l'edge corrispondente dal manifesto dell'applicazione dell'altra applicazione e ridistribuite.

Problema: l'output dell'applicazione non viene visualizzato nei log

Configura un logger Python su cui scrivere file di registro. `/opt/aws/panorama/logs` Questi vengono acquisiti in un flusso di log per il nodo contenitore del codice. I flussi di output e di errore standard vengono acquisiti in un flusso di registro separato chiamato`console-output`. Se lo utilizzi`print`, usa l'`flush=True`opzione per evitare che i messaggi rimangano bloccati nel buffer di output.

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

Fonte: AWS Panorama servizio

Ogni volta che si distribuisce una modifica a un'applicazione, si registra una versione di patch che rappresenta la configurazione del pacchetto e i file di asset per ogni pacchetto utilizzato. Utilizzate lo script cleanup patches per annullare la registrazione delle versioni di patch non utilizzate.

Stream da videocamera

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

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

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

Fonte: registro del nodo della fotocamera

L'appliance non riesce a connettersi allo stream della videocamera dell'applicazione. Quando ciò accade, l'uscita video è vuota o si blocca sull'ultimo fotogramma elaborato mentre l'applicazione attende un fotogramma di video dall'AWS PanoramaApplication SDK. Il software dell'appliance tenta di connettersi allo stream della telecamera e registra gli errori di timeout nel registro del nodo della telecamera. Verificate che l'URL dello stream della videocamera sia corretto e che il traffico RTSP sia instradabile tra la telecamera e l'appliance all'interno della rete. Per ulteriori informazioni, consulta Connessione di AWS Panorama Appliance alla rete.

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

Fonte: registro OCC

Il AWS Secrets Manager segreto con le credenziali dello stream della videocamera non è stato trovato. Eliminate lo stream della videocamera e ricreatelo.

Errore: Camera did not provide an H264 encoded stream

Fonte: registro del nodo della fotocamera

Lo stream della telecamera ha una codifica diversa da H.264, ad esempio H.265. Ridistribuite l'applicazione con uno stream di videocamera H.264. Per informazioni dettagliate sulle fotocamere supportate, vedere. Fotocamere supportate

Sicurezza AWS Panorama

Per AWS, la sicurezza del cloud ha la massima priorità. In quanto cliente AWS, è possibile trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle organizzazioni più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra te e AWS. Il modello di responsabilità condivisa descrive questo come sicurezza del cloud e sicurezza nel cloud:

- La sicurezza del cloud:AWS è responsabile della protezione dell'infrastruttura che esegue AWS i servizi nel AWS Cloud. AWS fornisce, inoltre, servizi utilizzabili in modo sicuro. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei programmi di conformità AWS. Per ulteriori informazioni sui programmi di conformità che si applicano ad AWS Panorama, consultaAWSServizi inclusi dal programma di conformità.
- Sicurezza nel cloud: la tua responsabilità è determinata dal servizio AWS che utilizzi. L'utente è anche responsabile per altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Questa documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si usa AWS Panorama. Nei seguenti argomenti viene illustrato come configurare AWS Panorama per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare gli altri servizi AWS che consentono di monitorare e proteggere le risorse AWS Panorama.

Argomenti

Funzionalità di sicurezza di AWS Panorama

Per proteggereapplicazioni, modellie hardware contro codice dannoso e altri exploit, AWS Panorama Appliance implementa una vasta gamma di funzionalità di sicurezza. Questi includono, tra l'altro, quanto segue.

- Crittografia disco completo— L'appliance implementa la crittografia completa di Linux Unified Key Setup (LUKS2). Tutti i dati del software di sistema e delle applicazioni sono crittografati con una chiave specifica per il tuo dispositivo. Anche con l'accesso fisico al dispositivo, un utente malintenzionato non può ispezionare il contenuto del suo spazio di archiviazione.
- Randomizzazione del layout di memoria— Per proteggersi dagli attacchi che indirizzano il codice eseguibile caricato nella memoria, AWS Panorama Appliance utilizza la randomizzazione del layout dello spazio degli indirizzi (ASLR). ASLR randomizza la posizione del codice del sistema operativo mentre viene caricato in memoria. Ciò impedisce l'uso di exploit che tentano di sovrascrivere o eseguire sezioni specifiche di codice prevedendo dove è memorizzato in fase di esecuzione.
- ambiente di esecuzione affidabile— L'appliance utilizza un ambiente TEE (Trusted Execution Environment) basato su ARM TrustZone, con risorse di storage, memoria ed elaborazione isolate. Le chiavi e altri dati sensibili memorizzati nella zona di attendibilità sono accessibili solo da un'applicazione attendibile, che viene eseguita in un sistema operativo separato all'interno del TEE. Il software AWS Panorama Appliance viene eseguito nell'ambiente Linux non affidabile insieme al codice dell'applicazione. Può accedere alle operazioni crittografiche solo inviando una richiesta all'applicazione sicura.
- Provisioning sicuro— Quando si effettua il provisioning di un'appliance, le credenziali (chiavi, certificati e altro materiale crittografico) trasferite sul dispositivo sono valide solo per un breve periodo. L'appliance utilizza le credenziali di breve durata per connettersiAWS IoTe richiede un certificato valido per un periodo più lungo. Il servizio AWS Panorama genera credenziali e le crittografa con una chiave hardcoded sul dispositivo. Solo il dispositivo che ha richiesto il certificato può decifrarlo e comunicare con AWS Panorama.
- Avvio sicuro— Quando il dispositivo si avvia, ogni componente software viene autenticato prima dell'esecuzione. La ROM di avvio, software hardcoded nel processore che non può essere modificata, utilizza una chiave di crittografia hardcoded per decrittografare il bootloader, che convalida il kernel dell'ambiente di esecuzione attendibile e così via.
- Kernel firmato— I moduli del kernel sono firmati con una chiave di crittografia asimmetrica. Il kernel del sistema operativo decrittografa la firma con la chiave pubblica e verifica che corrisponda alla firma del modulo prima di caricare il modulo in memoria.

- dm-verity— Analogamente a come vengono convalidati i moduli del kernel, l'appliance utilizza Linux Device Mapper`dm-verity`per verificare l'integrità dell'immagine software dell'appliance prima di installarla. Se il software dell'appliance viene modificato, non verrà eseguito.
- Prevenzione di rollback— Quando si aggiorna il software dell'appliance, l'apparecchio emette un fusibile elettronico sul SoC (sistema su chip). Ogni versione del software prevede che un numero crescente di fusibili venga bruciato e non può essere eseguito se ne soffiano altri.

Best practice relative alla sicurezza di AWS Panorama

Tieni presente le seguenti best practice quando usi l'appliance AWS Panorama.

- Fissare fisicamente l'apparecchio— Installare l'appliance in un server rack chiuso o in una stanza sicura. Limita l'accesso fisico al dispositivo al personale autorizzato.
- Protezione della connessione di rete dell'appliance— Connect l'appliance a un router che limita l'accesso alle risorse interne ed esterne. L'apparecchio deve connettersi alle telecamere, che possono trovarsi su una rete interna sicura. È inoltre necessario connettersi aAWS. Utilizzare la seconda porta Ethernet solo per la ridondanza fisica e configurare il router per consentire solo il traffico richiesto.

  Utilizza una delle configurazioni di rete consigliate per pianificare il layout di rete. Per ulteriori informazioni, consulta la pagina Connessione di AWS Panorama Appliance alla rete .
- Formattazione dell'unità USB— Dopo aver eseguito il provisioning di un accessorio, rimuovere l'unità USB e formattarla. L'appliance non utilizza l'unità USB dopo la registrazione con il servizio AWS Panorama. Formattare l'unità per rimuovere credenziali temporanee, file di configurazione e log di provisioning.
- Mantenere l'apparecchio aggiornato— Applicare tempestivamente gli aggiornamenti software dell'appliance. Quando visualizzi un'appliance nella console AWS Panorama, la console ti avvisa se è disponibile un aggiornamento software. Per ulteriori informazioni, consulta la pagina Gestione di un'appliance di AWS Panorama di AWS .

  Con laDescribeDeviceOperazione API, è possibile automatizzare il controllo degli aggiornamenti confrontando il`LatestSoftwareeCurrentSoftware`. Quando l'ultima versione del software è diversa dalla versione corrente, applicare l'aggiornamento con la console o utilizzando ilCrea lavoro per dispositivioperazione.
- Se si interrompe l'utilizzo di un apparecchio, ripristinarlo— Prima di spostare l'apparecchio fuori dal centro dati sicuro, ripristinarlo completamente. Con l'apparecchio spento e collegato, premere contemporaneamente il pulsante di accensione e reset per 5 secondi. In questo modo vengono eliminati le credenziali dell'account, le applicazioni e i registri dall'appliance.

  Per ulteriori informazioni, consulta la pagina Pulsanti e luci di AWS Panorama .
- Limita l'accesso a AWS Panorama e ad altri servizi AWS— IlAWSPanoramaFullAccessfornisce l'accesso a tutte le operazioni API di AWS Panorama e, se necessario, l'accesso ad altri servizi. Ove possibile, la politica limita l'accesso alle risorse in base alle convenzioni di denominazione. Ad

esempio, fornisce accesso aAWS Secrets Managersegreti che hanno nomi che inizianopanorama. Per gli utenti che necessitano di accesso in sola lettura o di accesso a un insieme di risorse più specifico, utilizzare il criterio gestito come punto di partenza per i criteri di privilegio minimo.

Per ulteriori informazioni, consultare Policy IAM basate su identità per AWS Panorama.

Protezione dei dati in AWS Panorama

Il modello di responsabilità AWS condivisa modello di di si applica alla protezione dei dati in AWS Panorama. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che esegue tutto l'Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Inoltre, sei responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS che utilizzi. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS.

Per garantire la protezione dei dati, ti suggeriamo di proteggere le credenziali Account AWS e di configurare singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Utilizza SSL/TLS per comunicare con le risorse AWS. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail.
- Utilizza le soluzioni di crittografia AWS, insieme a tutti i controlli di sicurezza predefiniti in Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se necessiti di moduli crittografici convalidati FIPS 140-2 quando accedi ad AWSattraverso un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con AWS Panorama o altro Servizi AWS utilizzando la console, l'API o AWS gli SDK. AWS CLI I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Sections

Crittografia in transito

Gli endpoint dell'API AWS Panorama supportano connessioni sicure solo tramite HTTPS. Quando gestisci le risorse AWS Panorama con l'AWS Management ConsoleSDK AWS o l'API AWS Panorama, tutte le comunicazioni vengono crittografate con Transport Layer Security (TLS). Anche la comunicazione tra AWS Panorama Appliance e AWS è crittografata con TLS. La comunicazione tra AWS Panorama Appliance e le telecamere tramite RTSP non è crittografata.

Per un elenco completo degli endpoint API, consulta Regioni ed endpoint AWS nel. Riferimenti generali di AWS

Appliance AWS Panorama

L'AWS Panorama Appliance dispone di porte fisiche per Ethernet, video HDMI e storage USB. Lo slot per schede SD, il Wi-Fi e il Bluetooth non sono utilizzabili. La porta USB viene utilizzata solo durante il provisioning per trasferire un archivio di configurazione all'appliance.

Il contenuto dell'archivio di configurazione, che include il certificato di provisioning dell'appliance e la configurazione di rete, non è crittografato. AWS Panorama non archivia questi file; possono essere recuperati solo quando registri un'appliance. Dopo aver trasferito l'archivio di configurazione su un'appliance, eliminalo dal computer e dal dispositivo di archiviazione USB.

L'intero file system dell'appliance è crittografato. Inoltre, l'appliance applica diverse protezioni a livello di sistema, tra cui la protezione dal rollback per gli aggiornamenti software richiesti, il kernel e il bootloader firmati e la verifica dell'integrità del software.

Quando smetti di usare l'appliance, esegui un ripristino completo per eliminare i dati dell'applicazione e reimpostare il software dell'appliance.

Applicazioni

Sei tu a controllare il codice da distribuire sul tuo dispositivo. Convalida tutto il codice dell'applicazione per verificare eventuali problemi di sicurezza prima di distribuirlo,

indipendentemente dalla sua origine. Se utilizzi librerie di terze parti nella tua applicazione, valuta attentamente le politiche di licenza e supporto per tali librerie.

L'utilizzo della CPU, della memoria e del disco dell'applicazione non è limitato dal software dell'appliance. Un'applicazione che utilizza troppe risorse può influire negativamente su altre applicazioni e sul funzionamento del dispositivo. Testa le applicazioni separatamente prima di combinarle o distribuirle in ambienti di produzione.

Gli asset applicativi (codici e modelli) non sono isolati dall'accesso all'interno dell'account, dell'appliance o dell'ambiente di compilazione. Le immagini dei container e gli archivi dei modelli generati dalla CLI dell'applicazione AWS Panorama non sono crittografati. Utilizza account separati per i carichi di lavoro di produzione e consenti l'accesso solo in base alle necessità.

Altri servizi

Per archiviare modelli e contenitori di applicazioni in modo sicuro in Amazon S3, AWS Panorama utilizza la crittografia lato server con una chiave gestita da Amazon S3. Per ulteriori informazioni, consulta la sezione Protezione dei dati mediante crittografia nella Guida per l'utente di Amazon Simple Storage Service.

Le credenziali dello streaming della telecamera sono crittografate quando sono archiviateAWS Secrets Manager. Il ruolo IAM dell'appliance le concede l'autorizzazione a recuperare il segreto per accedere al nome utente e alla password dello stream.

L'AWS Panorama Appliance invia i dati di log ad Amazon CloudWatch Logs. CloudWatch I log crittografano questi dati per impostazione predefinita e possono essere configurati per utilizzare una chiave gestita dal cliente. Per ulteriori informazioni, consulta Encrypt log data in CloudWatch Logs using AWS KMS nella Amazon CloudWatch Logs User Guide.

Gestione delle identità e degli accessi per AWS Panorama

AWS Identity and Access Management (IAM) è un Servizio AWSche consente agli amministratori di controllare in modo sicuro l'accesso alle risorse AWS. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse AWS Panorama. IAM è un Servizio AWS il cui uso non comporta costi aggiuntivi.

Argomenti

Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in AWS Panorama.

Utente del servizio: se utilizzi il servizio AWS Panorama per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzionalità di AWS Panorama per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in AWS Panorama, consultaRisoluzione dei problemi di identità e accesso ad AWS Panorama.

Amministratore del servizio: se sei responsabile delle risorse AWS Panorama presso la tua azienda, probabilmente hai pieno accesso ad AWS Panorama. Spetta a te determinare a quali funzionalità e risorse di AWS Panorama devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni

su come la tua azienda può utilizzare IAM con AWS Panorama, consultaCome funziona AWS Panorama con IAM.

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy per gestire l'accesso ad AWS Panorama. Per visualizzare esempi di policy basate sull'identità di AWS Panorama che puoi utilizzare in IAM, consulta. Esempi di policy basate sull'identità di AWS Panorama

Autenticazione con identità

L'autenticazione è la procedura di accesso ad AWS con le credenziali di identità. Devi essere autenticato (connesso a AWS) come utente root Utente root dell'account AWS, come utente IAM o assumere un ruolo IAM.

Puoi accedere ad AWS come identità federata utilizzando le credenziali fornite attraverso un'origine di identità. Gli utenti AWS IAM Identity Center (Centro identità IAM), l'autenticazione Single Sign-On (SSO) dell'azienda e le credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Se accedi ad AWS tramite la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere alla AWS Management Console o al portale di accesso AWS. Per ulteriori informazioni sull'accesso ad AWS, consulta la sezione Come accedere al tuo Account AWS nella Guida per l'utente di Accedi ad AWS.

Se accedi ad AWS in modo programmatico, AWS fornisce un Software Development Kit (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le richieste utilizzando le tue credenziali. Se non utilizzi gli strumenti AWS, devi firmare le richieste personalmente. Per ulteriori informazioni sulla firma delle richieste, consulta Firma delle richieste AWS nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. AWS consiglia ad esempio di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza dell'account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella Guida per l'utente di IAM.

Utente root di un Account AWS

Quando crei un Account AWS, inizi con una singola identità di accesso che ha accesso completo a tutti i Servizi AWS e le risorse nell'account. Tale identità è detta utente root Account AWS ed è

possibile accedervi con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente di IAM.

Utenti e gruppi IAM

Un utente IAM è una identità all'interno del tuo Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente di IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato Amministratori IAM e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Quando creare un utente IAM (invece di un ruolo) nella Guida per l'utente di IAM.

Ruoli IAM

Un ruolo IAM è un'identità all'interno di un Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. È possibile assumere temporaneamente un ruolo IAM nella AWS Management Console mediante lo scambio di ruoli. È possibile assumere un ruolo chiamando un'azione AWS CLI o API AWS oppure utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Creazione di un ruolo per un provider di identità di terza parte nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per ulteriori informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center.
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, per alcuni dei Servizi AWS, è possibile collegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.
- Accesso multi-servizio: alcuni Servizi AWS utilizzano funzionalità in altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Inoltro delle sessioni di accesso (FAS): quando si utilizza un utente o un ruolo IAM per eseguire operazioni in AWS, tale utente o ruolo viene considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che effettua la chiamata a un Servizio AWS, combinate con il Servizio AWS richiedente, per effettuare richieste a servizi a valle. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che necessita di interazioni con altri Servizi AWS o risorse per essere portata a termine. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM assunto da un servizio per eseguire operazioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.

- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati ai servizi sono visualizzati nell'account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: è possibile utilizzare un ruolo IAM per gestire credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 che eseguono richieste di AWS CLIo dell'API AWS. Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un ruolo AWS a un'istanza EC2, affinché sia disponibile per tutte le relative applicazioni, puoi creare un profilo dell'istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta Quando creare un ruolo IAM (invece di un utente) nella Guida per l'utente di IAM.

Gestione dell'accesso con policy

Per controllare l'accesso a AWS è possibile creare policy e collegarle a identità o risorse AWS. Una policy è un oggetto in AWS che, quando associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste policy quando un principale IAM (utente, utente root o sessione ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle policy viene archiviata in AWSsotto forma di documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente di IAM.

Gli amministratori possono utilizzare le policy AWSJSON per specificare l'accesso ai diversi elementi. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. Successivamente l'amministratore può aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione

`iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dalla AWS Management Console, la AWS CLI o l'API AWS.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono incorporate direttamente in un singolo utente, gruppo o ruolo. Le policy gestite sono policy autonome che possono essere collegate a più utenti, gruppi e ruoli in Account AWS. Le policy gestite includono le policy gestite da AWS e le policy gestite dal cliente. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente di IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile allegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy gestite da AWS da IAM in una policy basata su risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3, AWS WAF e Amazon VPC sono esempi di servizi che supportano le ACL. Per maggiori informazioni sulle ACL, consulta Panoramica delle liste di controllo degli accessi (ACL) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta altri tipi di policy meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Policy di controllo dei servizi (SCP): le SCP sono policy JSON che specificano il numero massimo di autorizzazioni per un'organizzazione o unità organizzativa (OU) in AWS Organizations. AWS Organizationsè un servizio per il raggruppamento e la gestione centralizzata degli Account AWSmultipli di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. La SCP limita le autorizzazioni per le entità negli account membri, compreso ogni Utente root dell'account AWS. Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per informazioni su come AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella Guida per l'utente di IAM.

Come funziona AWS Panorama con IAM

Prima di utilizzare IAM per gestire l'accesso ad AWS Panorama, è necessario comprendere quali funzionalità IAM sono disponibili per l'uso con AWS Panorama. Per avere una visione di alto livello di come AWS Panorama e altri AWS servizi funzionano con IAM, consulta AWSi servizi che funzionano con IAM nella IAM User Guide.

Per una panoramica delle autorizzazioni, delle policy e dei ruoli utilizzati da AWS Panorama, consultaAutorizzazioni AWS Panorama.

Esempi di policy basate sull'identità di AWS Panorama

Per impostazione predefinita, gli utenti e i ruoli IAM non sono autorizzati a creare o modificare risorse AWS Panorama. Inoltre, non sono in grado di eseguire attività utilizzando la AWS Management Console, AWS CLI o un'API AWS. Un amministratore IAM deve creare policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse specificate di cui hanno bisogno. L'amministratore deve quindi allegare queste policy a utenti o IAM che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consultare Creazione di policy nella scheda JSON nella Guida per l'utente di IAM.

Argomenti

- Best practice delle policy
- Utilizzo della console AWS Panorama
- Consentire agli utenti di visualizzare le loro autorizzazioni

Best practice delle policy

Le policy basate sull'identità determinano se qualcuno può creare, accedere o eliminare risorse AWS Panorama nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l'Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Nozioni di base sulle policy gestite da AWSe passaggio alle autorizzazioni con privilegio minimo: per le informazioni di base su come concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite da AWSche concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo Account AWS. Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo

policy gestite dal cliente di AWSspecifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.

- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi inoltre utilizzare le condizioni per concedere l'accesso alle operazioni di servizio, ma solo se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer nella Guida per l'utente di IAM.
- Richiesta dell'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o utenti root nel tuo Account AWS, attiva MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console AWS Panorama

Per accedere alla console AWS Panorama, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse AWS Panorama nel tuo AWS account. Se crei una policy basata su identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti e ruoli IAM) associate a tale policy.

Per ulteriori informazioni, consultare Policy IAM basate su identità per AWS Panorama

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono allegate alla relativa identità utente. La policy include le autorizzazioni per completare questa azione sulla console o a livello di programmazione utilizzando la AWS CLIo l'API AWS.

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

AWSpolitiche gestite per AWS Panorama

Una policy gestita da AWS è una policy autonoma creata e amministrata da AWS. Le policy gestite da AWS sono progettate per fornire autorizzazioni per molti casi d'uso comuni in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Ricorda che le policy gestite da AWS potrebbero non concedere autorizzazioni con privilegi minimi per i tuoi casi d'uso specifici perché possono essere utilizzate da tutti i clienti AWS. Consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle policy gestite da AWS. Se AWS aggiorna le autorizzazioni definite in una policy gestita da AWS, l'aggiornamento riguarda tutte le identità principali (utenti, gruppi e ruoli) a cui è collegata la policy. È molto probabile che AWS aggiorni una policy gestita da AWS quando viene lanciato un nuovo Servizio AWS o nuove operazioni API diventano disponibili per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWS nella Guida per l'utente di IAM.

AWS Panorama fornisce le seguenti policy gestite. Per i contenuti completi e la cronologia delle modifiche di ciascuna policy, consulta le pagine collegate nella console IAM.

- AWSPanoramaFullAccess— Fornisce accesso completo ad AWS Panorama, ai punti di accesso AWS Panorama in Amazon S3, alle credenziali dell'appliance inAWS Secrets Managere registri dei dispositivi in AmazonCloudWatch. Include il permesso di creare unruolo collegato ai serviziper AWS Panorama.
- AWSPanoramaServiceLinkedRolePolicy— Consente ad AWS Panorama di gestire le risorse in AWS IoT, AWS Secrets Manager e AWS Panorama.
- AWSPanoramaApplianceServiceRolePolicy— Consente a un'appliance AWS Panorama di caricare i log suCloudWatche per ottenere oggetti dai punti di accesso Amazon S3 creati da AWS Panorama.

AWS Panorama si aggiorna aAWSpolitiche gestite

La tabella seguente descrive gli aggiornamenti delle policy gestite per AWS Panorama.

Modifica	Descrizione	Data
AWSPanoramaFullAccess: aggiornamento a una policy esistente	Sono state aggiunte autorizzazioni alla politica utente per consentire agli utenti di visualizzare i gruppi di log nelCloudWatchConsole di log.	2022-01-13
AWSPanoramaFullAccess: aggiornamento a una policy esistente	Sono state aggiunte autorizzazioni alla politica utente per consentire agli utenti di gestire AWS Panoramaruolo collegato ai servizie per accedere alle risorse di AWS Panorama in altri servizi tra cui IAM, Amazon S3,CloudWatche Secrets Manager.	2021-10-20
AWSPanoramaApplianceServiceRolePolicy: nuova policy	Nuova politica per il ruolo del servizio AWS Panorama Appliance	2021-10-20
AWSPanoramaServiceLinkedRolePolicy: nuova policy	Nuova politica per il ruolo collegato ai servizi di AWS Panorama.	2021-10-20
AWS Panorama ha iniziato a monitorare le modifiche	AWS Panorama ha iniziato a monitorare le modifiche relative alAWSpolitiche gestite.	2021-10-20

Utilizzo di ruoli collegati ai servizi per AWS Panorama

AWS Panorama utilizza ruoli collegati al servizio AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo di ruolo IAM univoco collegato direttamente a AWS Panorama. I ruoli collegati ai servizi sono definiti automaticamente da AWS Panorama e includono tutte le autorizzazioni richieste dal servizio per eseguire chiamate agli altri servizi AWS per tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di AWS Panorama perché non dovrai più aggiungere manualmente le autorizzazioni necessarie. AWS Panorama definisce le autorizzazioni dei relativi ruoli associati ai servizi e, salvo diversamente definito, AWS Panorama potrà assumere solo i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di AWS Panorama perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consultare Servizi AWS che funzionano con IAM e cercare i servizi che riportano Sì nella colonna Ruolo collegato ai servizi. Scegliere un link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Sezioni

Autorizzazioni del ruolo collegato ai servizi per AWS Panorama

AWS Panoramausa il ruolo collegato al servizio denominato .Ruolo di servizio AWS per AWS Panorama— Consente ad AWS Panorama di gestire le risorse in AWS IoT, AWS Secrets Manager e AWS Panorama.

Al fine di assumere il ruolo, il ruolo collegato ai servizi AWSServiceRoleForAWSPanorama considera attendibili i seguenti servizi:

- `panorama.amazonaws.com`

La policy delle autorizzazioni del ruolo consente ad AWS Panorama di eseguire le seguenti operazioni:

- MonitorareAWS Panoramarisorse

- Manage (Gestione)AWS IoTrisorse per ilAWS PanoramaAppliance
- Accesso aAWS Secrets Managersegreti per ottenere le credenziali della fotocamera

Per l'elenco completo delle autorizzazioni,visualizza la politica relativa ai ruoli di AWS Panorama Servicenella console IAM.

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per AWS Panorama

Non devi creare manualmente un ruolo collegato ai servizi. Quando si registra un apparecchio nelAWS Management Console, ilAWS CLI, o ilAWSAPI,AWS Panoramacrea il ruolo collegato ai servizi per te.

Se si elimina questo ruolo collegato ai servizi e quindi deve essere creato di nuovo, è possibile utilizzare lo stesso processo per ricreare il ruolo nell'account. Quando si registra un apparecchio,AWS Panoramacrea di nuovo il ruolo collegato ai servizi per te.

Modifica di un ruolo collegato ai servizi per AWS Panorama

AWS Panoramanon consente di modificare il ruolo collegato ai servizi AWSServiceRoleForAWSPanorama. Dopo aver creato un ruolo collegato ai servizi, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per AWS Panorama

Se non è più necessario utilizzare una caratteristica o un servizio che richiede un ruolo collegato ai servizi, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare manualmente.

Per eliminare ilAWS Panoramarisorse utilizzate dal AWSServiceRoleForAWSPanorama, utilizzare le procedure descritte nelle seguenti sezioni di questa guida.

- Eliminare versioni e applicazioni

- Annullamento della registrazione di un'appliance

> **Note**
>
> Se il servizio AWS Panorama utilizza tale ruolo quando tenti di eliminare le risorse, è possibile che l'eliminazione non abbia esito positivo. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare il ruolo collegato ai servizi AWSServiceRoleForAWSPanorama, usare la console IAM,AWS CLI, o ilAWSAPI. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi AWS Panorama

AWS Panorama supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consultaAWSRegioni ed endpoint.

Prevenzione del confused deputy tra servizi

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. In AWS, la rappresentazione cross-service può comportare il problema confused deputy. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio di chiamata) chiama un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti che ti aiutano a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle policy delle risorse per limitare le autorizzazioni che AWS Panorama fornisce un altro servizio alla risorsa. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` devono utilizzare lo stesso ID account nella stessa istruzione di policy.

Il valore di`aws:SourceArn`deve essere l'ARN di unAWS Panoramadispositivo.

Il modo più efficace per proteggersi dal problema confuso dei deputati è quello di utilizzare il`aws:SourceArn`chiave del contesto della condizione globale con l'ARN completo della risorsa.

Se non si conosce l'ARN completo della risorsa o si sta specificando più risorse, utilizzare il`aws:SourceArn`chiave di condizione del contesto globale con caratteri jolly (`*`) per le porzioni sconosciute dell'ARN. Ad esempio, `arn:aws:servicename::123456789012:*`.

Per istruzioni su come proteggere il ruolo di servizioAWS Panoramautilizza per dare il permesso alAWS PanoramaAppliance, vediGarantire il ruolo dell'appliance.

Risoluzione dei problemi di identità e accesso ad AWS Panorama

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere problemi comuni che potresti incontrare quando lavori con AWS Panorama e IAM.

Argomenti

- Non sono autorizzato a eseguire un'azione in AWS Panorama
- Non sono autorizzato a eseguire iam: PassRole
- Desidero consentire a persone esterne al mio AWS account di accedere alle mie risorse AWS Panorama

Non sono autorizzato a eseguire un'azione in AWS Panorama

Se la AWS Management Console indica che non hai l'autorizzazione a eseguire un'operazione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

Il seguente errore di esempio si verifica quando l'utente `mateojackson` IAM tenta di utilizzare la console per visualizzare i dettagli su un'appliance ma non dispone delle `panorama:DescribeAppliance` autorizzazioni.

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

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa `my-appliance` utilizzando l'operazione `panorama:DescribeAppliance`.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue policy devono essere aggiornate per consentirti di trasferire un ruolo ad AWS Panorama.

Alcuni Servizi AWSconsentono di trasmettere un ruolo esistente a tale servizio, invece di creare un nuovo ruolo di servizio o un ruolo collegato ai servizi. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in AWS Panorama. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

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

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Per ulteriore assistenza con l'accesso, contatta l'amministratore AWS. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Desidero consentire a persone esterne al mio AWS account di accedere alle mie risorse AWS Panorama

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS Panorama supporta queste funzionalità, consultaCome funziona AWS Panorama con IAM.
- Per informazioni su come garantire l'accesso alle risorse negli Account AWSche possiedi, consulta Fornire l'accesso a un utente IAM in un altro Account AWS in tuo possesso nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso alle risorse ad Account AWS di terze parti, consulta Fornire l'accesso agli Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente di IAM.

- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consultare Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

Convalida della conformità per AWS Panorama

Per sapere se il Servizio AWSè coperto da programmi di conformità specifici, consulta i Servizi AWScoperti dal programma di conformità e scegli il programma di conformità desiderato. Per informazioni generali, consulta Programmi per la conformità di AWS.

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Download di report in AWS Artifact.

La responsabilità di conformità durante l'utilizzo dei Servizi AWSè determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'azienda e dalle normative vigenti. Per semplificare il rispetto della conformità, AWSmette a disposizione le seguenti risorse:

- Guide Quick Start per la sicurezza e conformità: queste guide all'implementazione illustrano considerazioni relative all'architettura e forniscono la procedura per l'implementazione di ambienti di base su AWSincentrati sulla sicurezza e sulla conformità.
- Architetture per la sicurezza e la conformità HIPAA su Amazon Web Services: questo whitepaper descrive come le aziende possono utilizzare AWSper creare applicazioni conformi alla normativa HIPAA.

  > **Note**
  >
  > Non tutti i Servizi AWSsono conformi ai requisiti HIPAA. Per ulteriori informazioni, consulta la sezione Riferimenti sui servizi conformi ai requisiti HIPAA.

- Risorse per la conformità AWS: una raccolta di cartelle di lavoro e guide suddivise per settore e area geografica.
- AWSGuide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- Valutazione delle risorse con le regole nella Guida per gli sviluppatori di AWS Config: il servizio AWS Configvaluta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti.
- AWS Security Hub: questo Servizio AWSfornisce una visione completa dello stato di sicurezza all'interno di AWS. La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse

AWSe verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.

- AWS Audit Manager: questo Servizio AWSaiuta a verificare continuamente l'utilizzo di AWSper semplificare la gestione dei rischi e della conformità alle normative e agli standard di settore.

Considerazioni aggiuntive sulla presenza di persone

Di seguito sono riportate alcune best practice da considerare quando si utilizza AWS Panorama per scenari in cui potrebbero essere presenti persone:

- Assicurati di conoscere e rispettare tutte le leggi e i regolamenti applicabili al tuo caso d'uso. Ciò può includere leggi relative al posizionamento e al campo visivo delle videocamere, requisiti di avviso e segnaletica per il posizionamento e l'utilizzo delle videocamere e i diritti delle persone che possono essere presenti nei tuoi video, incluso il loro diritto alla privacy.
- Tieni in considerazione l'effetto delle videocamere sulle persone e sulla loro privacy. Oltre ai requisiti legali, valuta se sia opportuno inserire avvisi nelle aree in cui sono collocate le telecamere e se collocarle in piena vista e prive di occlusioni, in modo che le persone non siano sorprese di trovarsi davanti alla telecamera.
- Adottate politiche e procedure appropriate per il funzionamento delle telecamere e la revisione dei dati ottenuti dalle telecamere.
- Prendi in considerazione i controlli di accesso, le limitazioni di utilizzo e i periodi di conservazione appropriati per i dati ottenuti dalle tue telecamere.

Sicurezza dell'infrastruttura in AWS Panorama

In quanto servizio gestito, AWS Panorama è protetto daAWSsicurezza di rete globale. Per informazioni sui servizi di sicurezza AWS e su come AWS protegge l'infrastruttura, consulta la pagina Sicurezza del cloud AWS. Per progettare l'ambiente AWS utilizzando le best practice per la sicurezza dell'infrastruttura, consulta la pagina Protezione dell'infrastruttura nel Pilastro della sicurezza di AWS Well-Architected Framework.

Tu usiAWSchiamate API pubblicate per accedere ad AWS Panorama attraverso la rete. I clienti devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. In alternativa, è possibile utilizzare AWS Security Token Service (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

Implementazione di AWS Panorama Appliance nel tuo datacenter

L'AWS Panorama Appliance necessita dell'accesso a Internet per comunicare conAWSservizi. È inoltre necessario accedere alla rete interna di telecamere. È importante considerare attentamente la configurazione di rete e fornire a ciascun dispositivo solo l'accesso necessario. Fai attenzione se la tua configurazione consente ad AWS Panorama Appliance di fungere da ponte verso una rete di telecamere IP sensibili.

Sei responsabile di quanto segue:

- La sicurezza di rete fisica e logica di AWS Panorama Appliance.
- Gestisci in modo sicuro le telecamere collegate alla rete quando usi AWS Panorama Appliance.
- Mantenere aggiornati AWS Panorama Appliance e il software della fotocamera.
- Rispettare tutte le leggi o i regolamenti applicabili associati al contenuto dei video e delle immagini raccolti dai tuoi ambienti di produzione, compresi quelli relativi alla privacy.

AWS Panorama Appliance utilizza flussi di telecamere RTSP non crittografati. Per ulteriori informazioni sulla connessione di AWS Panorama Appliance alla rete, consultaConnessione di AWS Panorama Appliance alla rete. Per informazioni dettagliate sulla crittografia, vedereProtezione dei dati in AWS Panorama.

Software per l'ambiente runtime in AWS Panorama

AWS Panorama fornisce software che esegue il codice dell'applicazione in un ambiente basato su Ubuntu Linux su AWS Panorama Appliance. AWS Panorama è responsabile di mantenere aggiornato il software nell'immagine dell'appliance. AWS Panorama rilascia regolarmente aggiornamenti software, che puoi applicare tramiteutilizzo della console AWS Panorama.

È possibile utilizzare le librerie nel codice dell'applicazione installandole nell'applicazione`Dockerfile`. Per garantire la stabilità delle applicazioni tra le build, scegli una versione specifica di ciascuna libreria. Aggiorna regolarmente le tue dipendenze per risolvere i problemi di sicurezza.

Rilasci

La tabella seguente mostra quando le funzionalità e gli aggiornamenti software sono stati rilasciati per il AWS Panorama servizio, il software e la documentazione. Per assicurarsi di avere accesso a tutte le funzionalità, aggiorna l'AWS Panoramaappliance alla versione più recente del software. Per ulteriori informazioni su una versione, consultate l'argomento collegato.

Modifica	Descrizione	Data
Aggiornamento del software dell'appliance	La versione 7.0.13 è un aggiornamento principale che modifica il modo in cui l'applian ce gestisce gli aggiornam enti software. Se si limita la comunicazione di rete in uscita dall'appliance o la si connette a una sottorete VPC privata, è necessario consentir e l'accesso a endpoint e porte aggiuntivi prima di applicare l'aggiornamento. Per ulteriori informazioni, consulta il registro delle modifiche.	28 dicembre 2023
Aggiornamento del software dell'appliance	La versione 6.2.1 include correzioni di bug. Per ulteriori informazioni, consulta il registro delle modifiche.	6 settembre 2023
Aggiornamento del software dell'appliance	La versione 6.0.8 include correzioni di bug e miglioram enti della sicurezza. Per ulteriori informazioni, consulta il registro delle modifiche.	6 luglio 2023
Aggiornamento del software dell'appliance	La versione 5.1.7 include correzioni di bug e miglioram	31 marzo 2023

	enti nella gestione degli errori. Per ulteriori informazioni, consulta il registro delle modifiche.	
Aggiornamento della console	È ora possibile acquistare l'AWS Panoramaappliance dalla console di gestione. Per concedere a un utente l'autorizzazione all'acquisto di dispositivi, consulta le politiche IAM basate sull'identità per AWS Panorama.	2 febbraio 2023
Aggiornamento del software dell'appliance	La versione 5.0.74 include correzioni di bug e miglioramenti nella gestione degli errori. Per ulteriori informazioni, consulta il registro delle modifiche.	23 gennaio 2023
Aggiornamento dell'API	È stata aggiunta `AllowMajorVersionUpdate` l'opzione `OTAJobConfig` per attivare gli aggiornamenti delle versioni principali del software dell'appliance. Per ulteriori informazioni, vedere. CreateJobForDevices	19 gennaio 2023

Nuovo strumento per gli sviluppatori	Un nuovo strumento, «sideloading», è disponibile nell' GitHub archivio degli AWS Panorama esempi. È possibile utilizzare questo strumento per aggiornare il codice dell'applicazione senza creare e distribuire un contenitore. Per ulteriori informazioni, consulta il file README.	16 novembre 2022
Aggiornamento dell'immagine di base dell'applicazione	La versione 1.2.0 aggiunge un'opzione di timeout`video_in.get()` , imposta la variabile di `AWS_REGION` ambiente e migliora la gestione degli errori. Per ulteriori informazioni, consulta il registro delle modifiche.	16 novembre 2022
Aggiornamento del software dell'appliance	La versione 5.0.42 include correzioni di bug e aggiornamenti di sicurezza. Per ulteriori informazioni, consulta il registro delle modifiche.	16 novembre 2022
Aggiornamento del software dell'appliance	La versione 5.0.7 aggiunge il supporto per il riavvio dei dispositivi in remoto e la sospensione dello streaming della videocamera da remoto. Per ulteriori informazioni, consultate il registro delle modifiche.	13 ottobre 2022

Aggiornamento del software dell'appliance	La versione 4.3.93 aggiunge il supporto per il recupero dei log da un dispositivo offline. Per ulteriori informazioni, consulta il registro delle modifiche.	24 agosto 2022
Aggiornamento del software dell'appliance	La versione 4.3.72 include correzioni di bug e aggiornamenti di sicurezza. Per ulteriori informazioni, consulta il registro delle modifiche.	23 giugno 2022
Supporto AWS PrivateLink	AWS Panoramasupporta gli endpoint VPC per la gestione AWS Panorama delle risorse da una sottorete privata. Per ulteriori informazioni, consulta Utilizzo degli endpoint VPC.	2 giugno 2022
Aggiornamento del software dell'appliance	La versione 4.3.55 migliora l'utilizzo dello storage per il registro. `console_output` Per ulteriori informazioni, consulta il registro delle modifiche.	5 maggio 2022
Lenovo ThinkEdge® SE70	Un nuovo dispositivo per AWS Panorama è disponibile presso Lenovo. Il Lenovo ThinkEdge® SE70, basato su Nvidia Jetson Xavier NX, supporta le stesse funzionalità dell'appliance. AWS Panorama Per ulteriori informazioni, consulta Dispositivi compatibili.	6 aprile 2022

Aggiornamento dell'immagine di base dell'applicazione	La versione 1.1.0 migliora le prestazioni durante l'esecuzione di thread in background e aggiunge un flag (is_cached) agli oggetti multimediali che indica se l'immagine è nuova. Per ulteriori informazioni, vedete gallery.ecr.aws.	29 marzo 2022
Aggiornamento del software dell'appliance	La versione 4.3.45 aggiunge il supporto per l'accesso alla GPU e le porte in ingresso. Per ulteriori informazioni, consulta il registro delle modifiche.	24 marzo 2022
Aggiornamento del software dell'appliance	La versione 4.3.35 migliora la sicurezza e le prestazioni. Per ulteriori informazioni, consulta il registro delle modifiche.	22 febbraio 2022
Politiche gestite aggiornate	AWS Identity and Access Managementle politiche gestite per sono AWS Panorama state aggiornate. Per i dettagli, consulta le policy gestite da AWS.	13 gennaio 2022
Registri di provisioning	Con il software 4.3.23, l'appliance scrive i registri su un'unità USB durante il provisioning. Per ulteriori informazioni, vedere Logs.	13 gennaio 2022

Configurazione del server NTP	È ora possibile configurare l'AWS Panoramaappliance per utilizzare un server NTP specifico per la sincroniz zazione dell'orologio. Configurare le impostazioni NTP durante la configura zione dell'appliance con altre impostazioni di rete. Per ulteriori informazioni, vedere Configurazione.	13 gennaio 2022
Regioni aggiuntive	AWS Panoramaè ora disponibi le nelle regioni Asia Pacifico (Singapore) e Asia Pacifico (Sydney).	13 gennaio 2022
Aggiornamento del software dell'appliance	La versione 4.3.4 aggiunge il supporto per l'`precision Mode` impostazione dei modelli e aggiorna il comportamento di registraz ione. Per ulteriori informazi oni, consulta il registro delle modifiche.	8 novembre 2021
Politiche gestite aggiornate	AWS Identity and Access Managementle politiche gestite per sono AWS Panorama state aggiornate. Per i dettagli, consulta le policy gestite da AWS.	20 ottobre 2021

Disponibilità generale	AWS Panoramaè ora disponibile per tutti i clienti nelle regioni Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda) e Canada (Centrale). Per acquistare un AWS Panorama elettrodomestico, visita. AWS Panorama	20 ottobre 2021
Anteprima	AWS Panoramaè disponibile su invito nelle regioni Stati Uniti orientali (Virginia settentrionale) e Stati Uniti occidentali (Oregon).	1 dicembre 2020

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.