Guida per gli sviluppatori

AWS Panorama

AWS Panorama: Guida per gli sviluppatori

Table of Contents

Avviso di fine del supporto: il 31 maggio 2026, AWS terminerà il supporto per AWS Panorama. Dopo il 31 maggio 2026, non potrai più accedere alla AWS Panorama console o AWS Panorama alle risorse. Per ulteriori informazioni, consulta AWS Panorama Fine del supporto.

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.

Che cos'è AWS Panorama?

AWS Panorama è un servizio che porta la visione artificiale nella rete di telecamere locali. L' AWS Panorama appliance o un altro dispositivo compatibile viene installato nel datacenter, lo si registra e si distribuiscono applicazioni di visione artificiale dal cloud. AWS Panorama AWS Panorama funziona con le telecamere di rete RTSP (Real Time Streaming Protocol) esistenti. L'appliance esegue applicazioni di visione artificiale sicure fornite dai AWS partner o applicazioni create dall'utente con l' AWS Panorama Application SDK.

L' AWS Panorama appliance è un'appliance edge compatta che utilizza un potente system-on-module (SOM) ottimizzato per carichi di lavoro di machine learning. L'appliance può eseguire più modelli di visione artificiale su più flussi video in parallelo 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).

L' AWS Panorama appliance consente di eseguire applicazioni di visione artificiale autonome sull'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. Grazie all'integrazione con altri servizi AWS, puoi AWS Panorama eseguire le seguenti operazioni:

- Analizza i modelli di traffico: utilizza l'SDK AWS per registrare dati per l'analisi della vendita al dettaglio in Amazon DynamoDB. Utilizza un'applicazione serverless per analizzare i dati raccolti nel tempo, rilevare anomalie nei dati e prevedere il comportamento futuro.
- Ricevi avvisi sulla sicurezza del sito: monitora le aree vietate in un sito industriale. Quando l'applicazione rileva una situazione potenzialmente pericolosa, carica un'immagine su Amazon Simple Storage Service (Amazon S3) e invia una notifica a un argomento di Amazon Simple Notification Service (Amazon SNS) in modo che i destinatari possano intraprendere azioni correttive.
- Migliora il controllo della qualità: monitora l'output di una linea di assemblaggio per identificare le parti non conformi ai requisiti. Evidenzia le immagini delle parti non conformi con testo e un riquadro di selezione e visualizzale su un monitor per essere esaminate dal team di controllo qualità.
- Raccogli dati di addestramento e test: carica immagini di oggetti che il tuo modello di visione artificiale non è in grado di identificare o per i quali la fiducia del modello nelle sue ipotesi era al limite. Utilizza un'applicazione serverless per creare una coda di immagini che devono essere etichettate. Etichetta le immagini e usale per riqualificare il modello in Amazon SageMaker AI.

AWS Panorama utilizza altri servizi AWS per gestire l' AWS Panorama appliance, accedere a modelli e codice e distribuire applicazioni. AWS Panorama fa il più possibile senza richiedere l'interazione con altri servizi, ma la conoscenza dei seguenti servizi può aiutarti a capire come funziona AWS Panorama .

- SageMaker IA: puoi utilizzare l' SageMaker intelligenza artificiale per raccogliere dati di addestramento da telecamere o sensori, creare un modello di apprendimento automatico e addestrarlo per la visione artificiale. AWS Panorama utilizza SageMaker AI Neo per ottimizzare i modelli da eseguire sull' AWS Panorama Appliance.
- Amazon S3: utilizzi i punti di accesso Amazon S3 per organizzare il codice dell'applicazione, i modelli e i file di configurazione per la distribuzione su un dispositivo. AWS Panorama
- AWS IoT— AWS Panorama utilizza AWS IoT servizi per monitorare lo stato dell' AWS Panorama appliance, gestire gli aggiornamenti software e distribuire applicazioni. Non è necessario AWS IoT utilizzarlo direttamente.

Per iniziare a utilizzare l' AWS Panorama appliance e saperne di più sul servizio, continua aIniziare con AWS Panorama.

Fine del supporto per AWS Panorama

Dopo un'attenta valutazione, abbiamo deciso di interrompere il supporto per AWS Panorama a partire dal 31 maggio 2026. AWS Panorama non accetterà più nuovi clienti a partire dal 20 maggio 2025. In qualità di cliente esistente con un account registrato al servizio prima del 20 maggio 2025, puoi continuare a utilizzare le funzionalità di AWS Panorama. Dopo il 31 maggio 2026, non sarà più possibile utilizzare AWS Panorama.

Alternative a AWS Panorama

Se sei interessato a un'alternativa ad AWS Panorama, AWS offre opzioni sia per gli acquirenti che per i costruttori.

Per una out-of-the-box soluzione, l'AWS Partner Network offre soluzioni di più partner. Puoi sfogliare le soluzioni nella libreria di soluzioni AWS di molti dei nostri partner. Queste soluzioni partner includono opzioni per applicazioni hardware, software, software as a service (SaaS), soluzioni gestite o implementazioni personalizzate in base alle tue esigenze. Questo approccio offre una soluzione adatta al vostro caso d'uso senza che dobbiate avere esperienza nella visione artificiale,

nell'intelligenza artificiale o nello sviluppo di applicazioni. Ciò consente in genere di velocizzare il time-to-value sfruttando le competenze specialistiche dei partner AWS.

Se preferisci creare la tua soluzione, AWS offre strumenti e servizi di intelligenza artificiale per aiutarti a sviluppare un'applicazione di visione artificiale basata sull'intelligenza artificiale e a gestire le applicazioni e i dispositivi periferici. Amazon SageMaker fornisce un set di strumenti per creare, addestrare e distribuire modelli di machine learning per i tuoi casi d'uso con infrastruttura, strumenti e flussi di lavoro completamente gestiti. Oltre a consentirti di creare i tuoi modelli, Amazon SageMaker JumpStart offre algoritmi di visione artificiale integrati che possono essere ottimizzati in base al tuo caso d'uso specifico.

Per la gestione di dispositivi e applicazioni all'edge, AWS IoT Greengrass è una soluzione collaudata e sicura per distribuire e aggiornare applicazioni per dispositivi IoT. Per un'implementazione basata su server, AWS Systems Manager fornisce una suite di strumenti per la gestione dei server e Amazon EKS Anywhere o ECS Anywhere possono gestire i contenitori di applicazioni sui server edge. Amazon fornisce alcune linee guida per la gestione dei dispositivi edge, insieme a risorse aggiuntive nella Sezione 4 del white paper Securing Internet of Things (IoT) with AWS. Questo approccio costruttivo ti fornisce gli strumenti per accelerare lo sviluppo dell'intelligenza artificiale e della gestione dei dispositivi, fornendo al contempo la flessibilità completa per creare una soluzione che soddisfi i tuoi requisiti esatti e si integri con l'infrastruttura hardware e software esistente. Ciò consente in genere di ridurre i costi operativi di una soluzione.

Migrazione da AWS Panorama

Per spostare un'applicazione esistente da AWS Panorama a un'implementazione alternativa, dovrai sostituire il dispositivo hardware esistente, migrare l'applicazione dal servizio AWS Panorama e implementare la gestione e la sicurezza dell'edge per la nuova soluzione. Ciascuna di queste aree verrà esplorata in dettaglio di seguito:

Sostituzione dell'hardware

L'appliance AWS Panorama esistente è basata sulla piattaforma Nvidia Jetson Xavier. L'hardware può essere sostituito con un off-the-shelf dispositivo simile basato sulla piattaforma Nvidia Jetson di ultima generazione che soddisfa i requisiti richiesti, oppure con un server edge. Sebbene la maggior parte delle implementazioni di AWS Panorama possa essere sostituita con un dispositivo simile, abbiamo riscontrato che alcuni clienti che utilizzano un gran numero di telecamere in un'unica posizione hanno scoperto che un server è un'alternativa migliore.

Migrazione delle applicazioni

Le applicazioni AWS Panorama devono essere riscritte per eliminare l'uso di chiamate API specifiche di AWS Panorama. Le applicazioni AWS Panorama supportano solo l'input video tramite feed RTSP (Real-Time Streaming Protocol) utilizzando H.264 e tali ingressi video vengono forniti utilizzando nodi telecamera nell'SDK del dispositivo AWS Panorama.

Per eseguire il porting di un'applicazione esistente, è necessario implementare una classe di applicazioni simile a AWS Panorama in modo che il codice esistente possa essere riutilizzato per lo più. Il codice di esempio è disponibile nel file banner-code.zip che mostra un esempio di questa implementazione utilizzando sia PyAV che OpenCV.

Si tratta di un approccio semplice con una quantità minima di modifiche al codice, ma presenta molte delle stesse limitazioni dell'attuale implementazione basata su AWS Panorama in termini di tipi di flussi video supportati.

Un'altra opzione potrebbe essere quella di riprogettare l'applicazione per utilizzare meglio le risorse di sistema e supportare nuove funzionalità applicative. Per questa opzione, si utilizza GStreamero si DeepStreamimplementa la pipeline multimediale dalla sorgente multimediale ai risultati di inferenza e alla logica di business, oppure si utilizza un'implementazione di runtime di machine learning (ML) più ricca di funzionalità e con prestazioni migliori, come il server di inferenza Nvidia Triton. Questo approccio richiede modifiche a una parte maggiore della pipeline di elaborazione video, ma è al contempo più efficiente e consente una maggiore flessibilità per supportare una gamma più ampia di codec, tipi di telecamere e altri sensori.

Gestione e sicurezza dei dispositivi perimetrali

Indipendentemente dalla pipeline multimediale, dovrai anche implementare un archivio sicuro per le credenziali, ad esempio nome utente e password dello stream RTSP. AWS offre diversi modi per archiviare in modo sicuro i parametri per le applicazioni:

- Il servizio AWS IoT Device Shadow viene utilizzato per archiviare i parametri che vengono passati alle applicazioni e per tracciare lo stato delle applicazioni sul dispositivo perimetrale.
- AWS Secrets Manager viene utilizzato per archiviare tali credenziali per proteggere meglio le credenziali di accesso ai flussi multimediali.
- Se utilizzi Amazon EKS o Amazon ECS, puoi anche utilizzare l'archivio di parametri sicuro di AWS System Manager per credenziali e altri parametri dell'applicazione.

La scelta dipende dai requisiti di sicurezza dell'applicazione e dagli altri AWS prodotti che intendi utilizzare per implementare l'applicazione.

Quando sostituisci l'appliance AWS Panorama con un dispositivo edge generico, devi anche implementare le funzionalità di sicurezza richieste per le tue applicazioni e configurare i dispositivi in modo che soddisfino i tuoi requisiti di sicurezza. AWS fornisce indicazioni in merito nel Security Pillar di AWS Well-Architected Framework. Sebbene il framework si concentri principalmente sulle applicazioni cloud, la maggior parte dei principi si applica anche ai dispositivi edge. Inoltre, è necessario utilizzare le funzionalità di sicurezza hardware della soluzione scelta, come l'integrazione della sicurezza hardware AWS IoT Greengrass V2, e utilizzare le funzionalità di sicurezza fornite dal sistema operativo e/o dal dispositivo scelto, come la crittografia completa del disco.

Riepilogo

Sebbene AWS Panorama preveda di chiudere il 31 maggio 2026, AWS offre un potente set di servizi e soluzioni AI/ML sotto forma di SageMaker strumenti Amazon per creare modelli di visione artificiale e servizi di gestione dei dispositivi, come AWS IoT Greengrass, Amazon EKS e Amazon ECS Anywhere e AWS System Manager per supportare lo sviluppo di soluzioni simili. AWS offre anche una gamma di offerte di partner dell'AWS Partner Network se preferisci acquistare invece di creare una soluzione. Vengono forniti esempi di codice e linee guida all'implementazione per aiutarti a migrare verso una soluzione alternativa, se lo desideri. È consigliabile esplorare queste opzioni per determinare quella più adatta alle proprie esigenze specifiche.

Per ulteriori dettagli, consulta le seguenti risorse:

- Amazon SageMaker Developer Guide — Documentazione dettagliata su come creare un modello o lavorare con algoritmi di visione artificiale integrati disponibili in SageMaker JumpStart.
- AWS IoT Core Developer Guide: documentazione dettagliata su come connettere e gestire i dispositivi IoT.
- AWS IoT Greengrass V2 Developer Guide — Documentazione dettagliata su come creare, distribuire e gestire applicazioni IoT sui tuoi dispositivi.
- Guida per sviluppatori ECS Anywhere - Documentazione dettagliata sull'esecuzione di ECS all'edge.
- Guida alle migliori pratiche di EKS Anywhere - Documentazione dettagliata sull'utilizzo di EKS all'edge.
- Libreria di soluzioni AWS: offerte dei partner di una vasta gamma di fornitori che offrono soluzioni di visione artificiale predefinite o personalizzate.

- Panorama FAQs - Informazioni panoramiche aggiuntive.

Domande frequenti

Quali sono le tempistiche per l'interruzione della produzione di Panorama?

L'annuncio è stato dato il 20 maggio 2025. Dopo questa data, i clienti che non sono attivi sul servizio non avranno più accesso a Panorama. I clienti attivi potranno continuare a utilizzare il servizio normalmente fino al 31 maggio 2026. I clienti hanno tempo fino a quel momento per spostare la propria applicazione su una soluzione alternativa e migrare le applicazioni di Panorama. Dopo il 31 maggio 2026, qualsiasi applicazione che tenta di accedere al servizio Panorama non funzionerà più e i dispositivi Panorama non funzioneranno più.

In che modo verranno influenzati i clienti esistenti?

I clienti esistenti possono continuare a utilizzare il servizio normalmente fino al 31 maggio 2026. Dopodiché, le applicazioni che tentano di accedere a Panorama non funzioneranno più. Inoltre, i dispositivi Panorama non funzioneranno più dopo tale data.

I nuovi clienti vengono accettati?

No. A partire dal 20 maggio 2025, solo i clienti che sono utenti attivi di Panorama avranno accesso al servizio. Se un cliente dispone di applicazioni del servizio utilizzate in precedenza a cui deve accedere, può presentare una richiesta all'assistenza clienti per richiedere l'accesso al proprio account. Se un cliente non utilizza in precedenza il servizio, non gli verrà concesso l'accesso.

Quali sono le alternative che i clienti possono esplorare?

AWS offre una gamma di servizi che possono sostituire le funzionalità Panorama. Consigliamo ai clienti di utilizzare l' off-the-shelfhardware e gestire il dispositivo e l'applicazione tramite la combinazione di AWS IoT Core, AWS IoT Greengrass, Amazon AKS Anywhere, Amazon ECS Anywhere e/o AWS System Manager che soddisfi i loro requisiti. L'AWS Partner Network mette inoltre a disposizione diverse soluzioni offerte da partner con competenze specifiche in Computer Vision che i clienti possono prendere in considerazione.

In che modo i clienti possono migrare da Panorama?

Le applicazioni Panorama devono essere modificate per rimuovere eventuali dipendenze da Panorama-specific APIs, che riguardano principalmente la connessione e lo streaming della

telecamera. AWS ha fornito un codice di esempio per mostrare come apportare queste modifiche. Una volta rimosse tali dipendenze, l'applicazione può essere spostata su una piattaforma hardware alternativa.

Se dovessi riscontrare problemi a partire dal 20 maggio 2025, quale supporto sarà disponibile?

AWS continuerà a fornire supporto a Panorama fino alla fine del periodo di notifica della sospensione (31 maggio 2026). Per qualsiasi esigenza di supporto, i clienti devono inviare una richiesta di assistenza tramite i normali canali di assistenza. AWS fornirà aggiornamenti di sicurezza, correzioni di bug e miglioramenti della disponibilità.

Non posso effettuare la migrazione prima del 31 maggio 2026. La data può essere prorogata?

Siamo certi che le alternative disponibili per Panorama consentano ai clienti di migrare a una soluzione alternativa entro il 31 maggio 2026 e non abbiamo intenzione di estendere la disponibilità del servizio oltre tale data.

La mia applicazione edge continuerà a funzionare dopo la fine del servizio?

No. Il dispositivo e le applicazioni Panorama dipendono dalla connettività al servizio cloud Panorama. Una volta interrotto il servizio il 31 maggio 2026, né l'applicazione Panorama né il dispositivo Panorama continueranno a funzionare.

Iniziare con AWS Panorama

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

> **Note**
>
> Per acquistare un AWS Panorama dispositivo, accedi alla console. AWS Panorama

L'applicazione AWS Panorama di esempio dimostra l'uso delle funzionalità. AWS Panorama Include un modello che è stato addestrato con l' SageMaker intelligenza artificiale e un codice di esempio che utilizza l' AWS Panorama Application SDK per eseguire inferenze e generare video. L'applicazione di esempio include un 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 modelli e fotocamere e le specifiche hardware dell'appliance. AWS Panorama Se non avete ancora acquistato un dispositivo e delle fotocamere o avete intenzione di sviluppare modelli di visione artificiale personalizzati, consultate innanzitutto questi argomenti per ulteriori informazioni.

Argomenti

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. Puoi utilizzare l'AWS Panorama Application CLI 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. Utilizzi la console AWS Panorama per registrare un'appliance, aggiornarne il software e distribuirvi applicazioni. Il software su AWS Panorama Appliance si collega agli stream delle telecamere, invia frame di video all'applicazione e visualizza l'output video su un display collegato.

L'AWS Panorama Appliance è un dispositivo edge basato su Nvidia Jetson AGX Xavier. Invece di inviare immagini al AWS cloud 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 dell' AWS Panorama appliance.

Dispositivi compatibili

Oltre ad AWS Panorama Appliance, AWS Panorama supporta i dispositivi compatibili dei AWS partner. 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® SE7 0: 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 machine learning 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 la CLI dell'applicazione AWS Panorama. L'AWS Panorama Application CLI 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 delle AWS Panorama applicazioni.

Nodi

Un'applicazione comprende più componenti chiamati nodi, che rappresentano input, output, modelli e codice. Un nodo può essere solo configurato (input e output) o includere artefatti (modelli e codice). I nodi di codice di un'applicazione sono raggruppati in pacchetti di nodi caricati su un punto di accesso Amazon S3, a cui l'AWS Panorama Appliance può accedervi. Un manifesto dell'applicazione è un file di configurazione che definisce le connessioni tra i nodi.

Per ulteriori informazioni, consulta Nodi applicativi.

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 AI o nel tuo ambiente di sviluppo. Per ulteriori informazioni, consulta ???.

Configurazione di AWS Panorama Appliance

Per iniziare a utilizzare AWS Panorama Appliance 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.

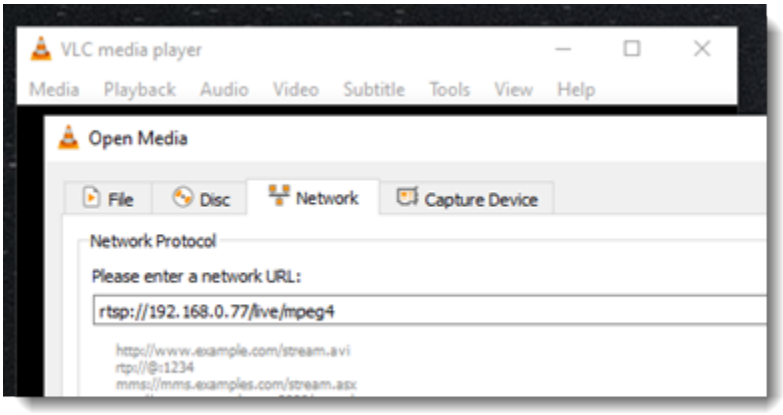
Sections

Prerequisiti

Per seguire questo tutorial, è necessario un dispositivo AWS Panorama o compatibile e il seguente hardware:

- Display: un display con ingresso HDMI per visualizzare l'output dell'applicazione di esempio.
- Unità USB (inclusa con AWS Panorama Appliance): un'unità di memoria flash USB 3.0 FAT32 formattata con almeno 1 GB di spazio di archiviazione, per il trasferimento di un archivio con file di configurazione e un certificato su AWS Panorama Appliance.
- Telecamera: 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 stream, aprendolo come sorgente 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'appliance 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 delle appliance e ai log delle appliance in Gestione dei segreti AWS 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 da AWS Panorama Appliance.

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

Registrazione e configurazione di AWS Panorama Appliance

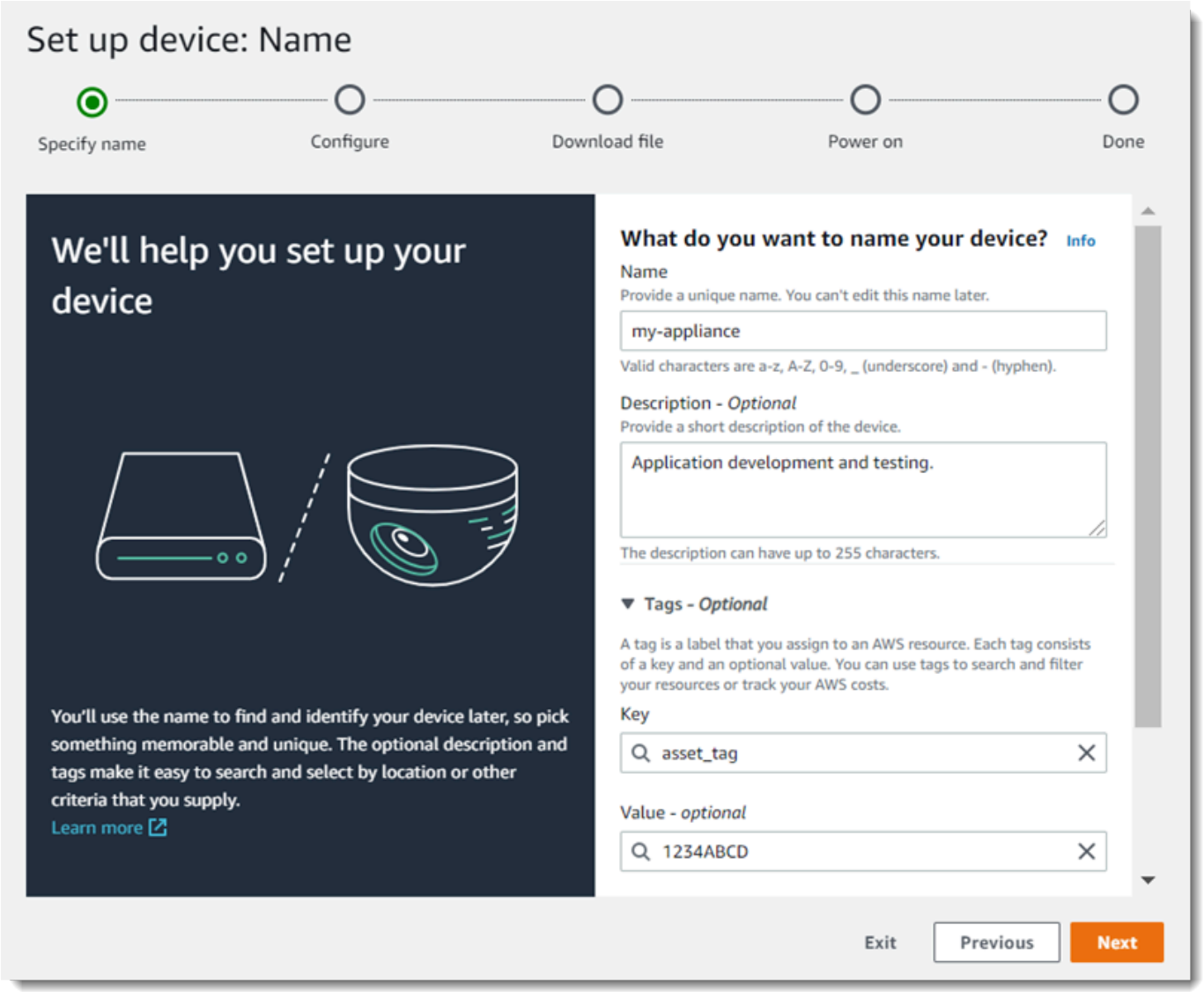
L'AWS Panorama Appliance è un dispositivo hardware che si collega a telecamere abilitate alla rete tramite una connessione di rete locale. Utilizza un sistema operativo basato su Linux che include l'SDK per applicazioni AWS Panorama e il software di supporto per l'esecuzione di applicazioni di visione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

Per connettersi AWS per la gestione e la distribuzione delle applicazioni, 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 approvvigionamento generato con questa procedura è valido solo per 5 minuti. Se non completate la procedura di registrazione entro questo lasso di tempo, dovete ricominciare da capo.

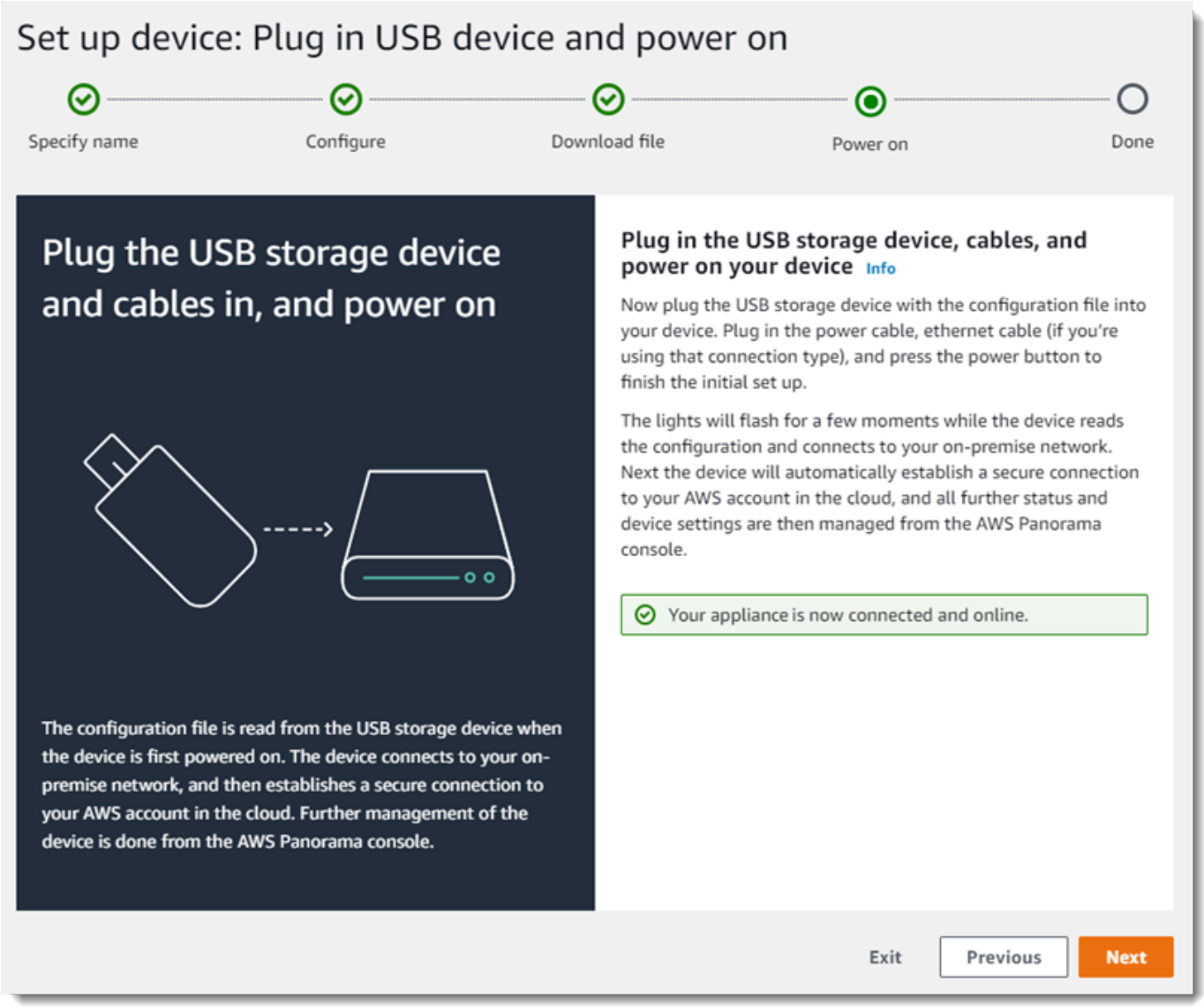
Per registrare un apparecchio

1. Connect l'unità USB al computer. Preparare l'apparecchiatura collegando i cavi di rete e di alimentazione. L'appliance si accende e attende che venga collegata un'unità USB.
2. Apri la pagina introduttiva della console AWS Panorama.
3. Scegli Aggiungi dispositivo.
4. Scegli Inizia la configurazione.
5. Inserisci un nome e una descrizione per la risorsa del dispositivo che rappresenta l'appliance in AWS Panorama. Seleziona Next (Successivo).



6. Se devi assegnare manualmente un indirizzo IP, un server NTP o impostazioni DNS, scegli Impostazioni di rete avanzate. Altrimenti, scegli Next (Successivo).
7. Scegli Scarica archivio. Scegli Next (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 di rete su se stesso e si connette al AWS Cloud. La spia di stato dell'appliance passa dal verde al blu mentre completa la connessione, e poi torna al verde.
10. Scegliere Next (Avanti) per continuare.



11. Seleziona Fatto.

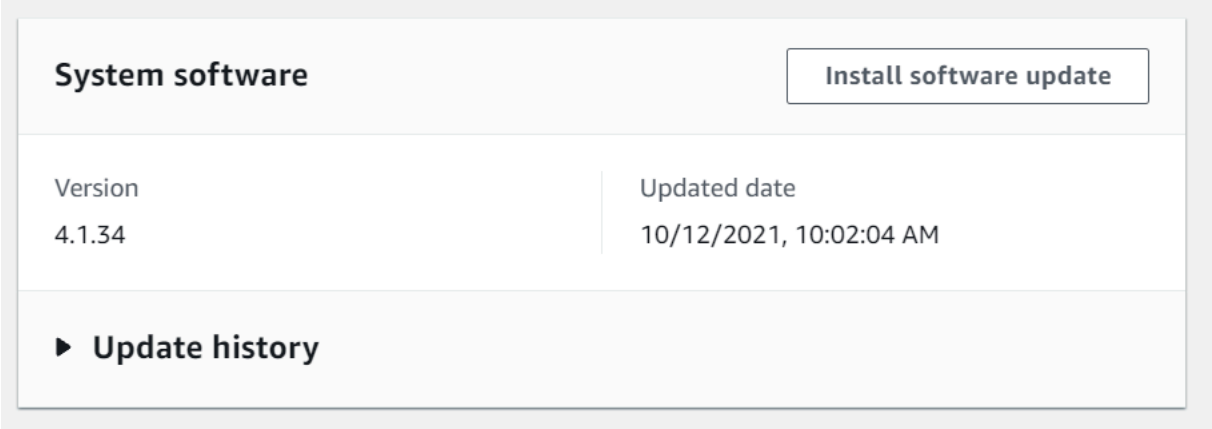
Aggiorna il software dell'appliance

AWS Panorama Appliance ha diversi componenti software, tra cui un sistema operativo Linux, l'SDK per applicazioni AWS Panorama e il supporto di librerie e framework di 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. Scegli un'appliance.
3. Scegli Impostazioni
4. In Software di sistema, scegli Installa l'aggiornamento del 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.

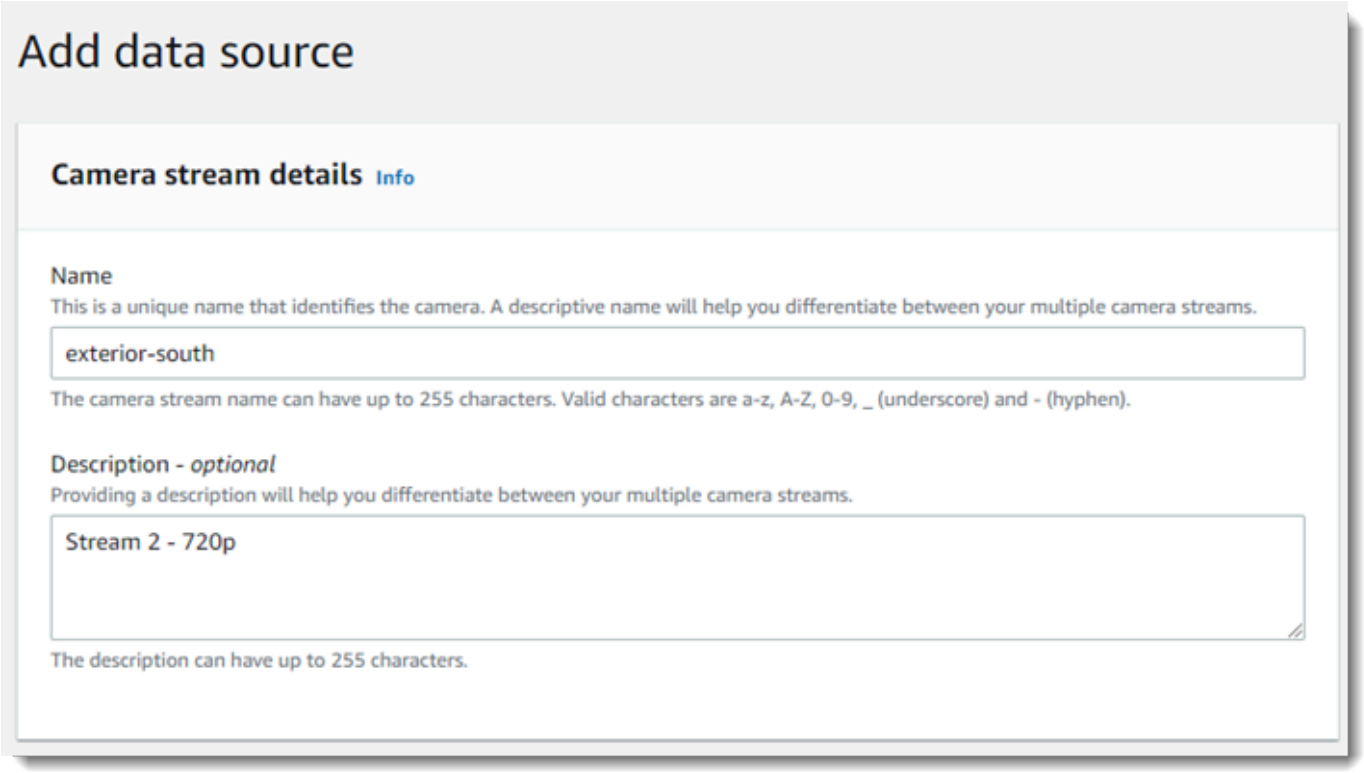
Il processo di aggiornamento può richiedere 30 minuti o più. Puoi monitorarne l'avanzamento nella console AWS Panorama o su un monitor connesso. Al termine del processo, l'appliance si riavvia.

Aggiungi uno stream da videocamera

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

Per registrare uno stream di telecamere

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



3. Configura le impostazioni seguenti.

   - Nome: un nome per lo stream della telecamera.
   - 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, specificate il nome utente e la password.
4. Seleziona Salva.

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

Passaggi successivi

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

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

Distribuzione dell'applicazione di esempio AWS Panorama

Dopo aver configurato la tua AWS Panorama Appliance o un dispositivo compatibile e aver aggiornato il relativo software, distribuisci un'applicazione di esempio. Nelle sezioni seguenti, importi un'applicazione di esempio con la CLI 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'SDK dell'applicazione AWS Panorama per caricare un modello, ottenere immagini ed eseguire il modello. L'applicazione sovrappone quindi i risultati al video originale e lo trasmette 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 del fabbisogno di personale durante le festività e altri eventi, misurare l'efficacia degli annunci pubblicitari e delle promozioni di vendita o ottimizzare il posizionamento degli espositori e la gestione dell'inventario.

Sections

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 codici, i comandi sono preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

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

Per i 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, consultaConfigurazione di un ambiente di sviluppo in Windows.

Utilizzi Python per sviluppare applicazioni AWS Panorama e installare strumenti con pip, il gestore di pacchetti di Python. Se non hai già Python, installa 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 Docker: Get Docker

Questo tutorial utilizza la CLI dell'applicazione AWS Panorama per importare l'applicazione di esempio, creare pacchetti e caricare artefatti. La CLI dell'applicazione AWS Panorama utilizza AWS Command Line Interface (AWS CLI) per chiamare le operazioni delle API di servizio. Se lo possiedi già AWS CLI, aggiornalo alla versione più recente. Per installare la CLI dell'applicazione AWS Panorama e utilizzare AWS CLI. `pip`

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

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

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

Importa l'applicazione di esempio

Per importare l'applicazione di esempio da utilizzare nel tuo account, utilizza la CLI 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, nella `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 dell'applicazione che viene eseguito sull'appliance, utilizzare il comando. `panorama-cli build-container`

```
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 la CLI dell'applicazione AWS Panorama consiste nel registrare il codice e i nodi del modello dell'applicazione e caricare gli asset su un punto di accesso Amazon S3 fornito dal servizio. Gli asset includono l'immagine del contenitore del codice, il modello e un file descrittore per ciascuno di essi. 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. Apri la pagina Applicazioni distribuite della console AWS Panorama.
2. Scegli Deploy application.
3. Incolla il contenuto del manifesto dell'applicazione nell'editor di testo. `graphs/aws-panorama-sample/graph.json` Scegli Next (Successivo).
4. Per Nome applicazione, immetti `aws-panorama-sample`.
5. Scegliete Procedi alla distribuzione.
6. Scegli Inizia la distribuzione.
7. Scegli Avanti senza selezionare un ruolo.
8. Scegli Seleziona dispositivo, quindi scegli il tuo dispositivo. Scegli Next (Successivo).
9. Nel passaggio Seleziona fonti di dati, scegli Visualizza input e aggiungi lo stream della videocamera come fonte di dati. Scegli Next (Successivo).

10. Nella fase di configurazione, scegli Avanti.
11. Scegli Distribuisci, quindi scegli Fine.
12. Nell'elenco delle applicazioni distribuite, scegli. aws-panorama-sample

Aggiorna questa pagina per gli aggiornamenti o usa lo script seguente 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 i log dell'applicazione e del dispositivo in Amazon CloudWatch Logs.

Visualizza l'output

Una volta completata la distribuzione, l'applicazione inizia a elaborare il flusso video e invia i log a CloudWatch.

Per visualizzare i log in Logs CloudWatch

1. Apri la pagina dei gruppi di log della console CloudWatch 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, collegate l'apparecchiatura 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 genera 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 multidimensionale. Per esempi di applicazioni con modelli più complessi, vedere. Applicazioni, script e modelli di esempio

Abilita l'SDK per Python

L'applicazione di esempio utilizza il AWS SDK per Python (Boto) per inviare i parametri ad Amazon CloudWatch. Per abilitare questa funzionalità, crea un ruolo che conceda all'applicazione l'autorizzazione a inviare metriche e ridistribuisci l'applicazione con il ruolo associato.

L'applicazione di esempio include un CloudFormation modello che crea un ruolo con le autorizzazioni necessarie. Per creare il ruolo, utilizzare il `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. Apri la pagina Applicazioni distribuite della console AWS Panorama.
2. Scegliere un'applicazione.
3. Scegliere Replace (Sostituisci).
4. Completa i passaggi per distribuire l'applicazione. Nel ruolo Specificare IAM, scegli il ruolo che hai creato. Il suo nome inizia con `aws-panorama-sample-runtime`.
5. Al termine dell'implementazione, apri la CloudWatchconsole e visualizza le metriche nel namespace. `AWSPanoramaApplication` Ogni 150 frame, l'applicazione registra e carica le metriche relative all'elaborazione dei frame e al tempo di inferenza.

Eliminazione

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

Per rimuovere l'applicazione dall'appliance

1. Apri la pagina Applicazioni distribuite della console AWS Panorama.

2. Scegliere un'applicazione.
3. Scegli Elimina dal dispositivo.

Passaggi successivi

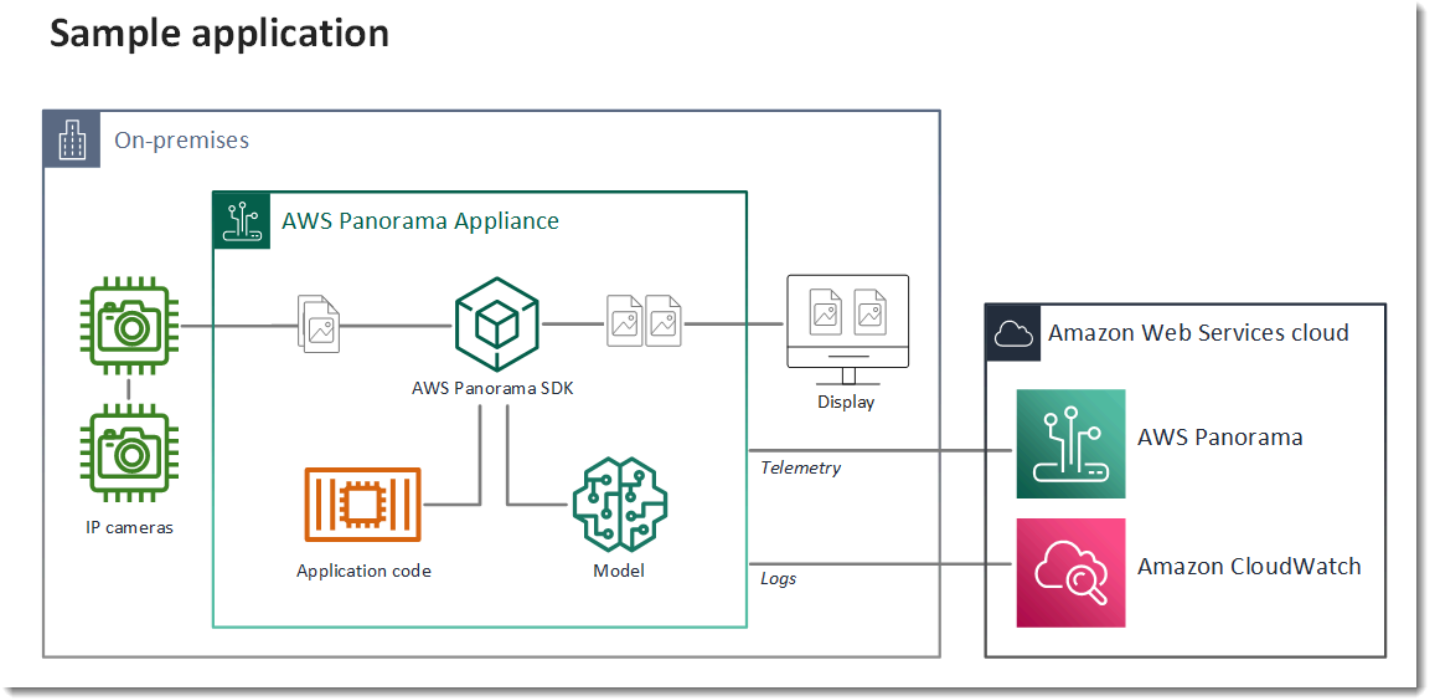
Se hai riscontrato errori durante la distribuzione o l'esecuzione dell'applicazione di esempio, consultaRisoluzione dei problemi.

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

Sviluppo di applicazioni AWS Panorama

Puoi utilizzare l'applicazione di esempio per conoscere la struttura dell'applicazione AWS Panorama e come punto di partenza per la tua applicazione.

Il diagramma seguente mostra i componenti principali dell'applicazione in esecuzione su un 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 trasmette video su uno schermo collegato ma non invia dati di 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 inviarli a un modello di visione artificiale che rileva gli oggetti. L'applicazione visualizza il risultato su un display HDMI collegato all'appliance.

Sections

- Passaggi successivi

Il manifesto dell'applicazione

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

I pacchetti sono codice, configurazione e file binari per il codice dell'applicazione, i modelli, le fotocamere e i display. L'applicazione 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. Il `abstract_rtsp_media_source` pacchetto è un segnaposto per una telecamera che sostituisci durante la distribuzione. Il `hdmi_data_sink` pacchetto rappresenta il connettore di uscita HDMI sul dispositivo.

I nodi sono interfacce per i pacchetti, nonché parametri non relativi ai pacchetti che possono avere valori predefiniti che possono essere sostituiti al momento della distribuzione. I pacchetti di codice e modello definiscono le interfacce nei `package.json` file che specificano input e output, che possono essere flussi video o un tipo di dati di base come float, booleano o stringa.

Ad esempio, il `code_node` nodo fa riferimento a un'interfaccia del pacchetto. `SAMPLE_CODE`

```
        "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 basato sulla logica aziendale e che richiede un flusso video denominato `video_in` e un numero a virgola mobile `threshold` denominato input. L'interfaccia specifica inoltre che il codice richiede un buffer di flusso video denominato `video_out` per trasmettere il 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"
                    }
                ]
            }
        ]
    }
}
```

Tornando al manifesto dell'applicazione, il `camera_node` nodo rappresenta un flusso video proveniente da una videocamera. Include un decoratore che appare nella console quando si distribuisce l'applicazione, e richiede di scegliere uno stream di videocamera.

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

```
            {
                "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 parametrico`threshold_param`, definisce il parametro della soglia di confidenza utilizzato dal codice dell'applicazione. Ha un valore predefinito di 60 e può essere sostituito durante la distribuzione.

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

```
            {
                "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`, crea connessioni tra i nodi. Il flusso video della telecamera e il parametro di soglia si collegano all'ingresso del nodo di codice e l'uscita video dal nodo di codice si collega 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 propria applicazione.

Il nome di ogni pacchetto deve essere unico nel tuo account. Se tu e un altro utente del tuo account utilizzate entrambi un nome di pacchetto generico come `code` o`model`, potreste ottenere la versione sbagliata del pacchetto durante la distribuzione. Cambia il nome del pacchetto di codice con uno che rappresenti la tua applicazione.

Per rinominare il pacchetto di codice

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

    - Manifesto dell'applicazione: `graphs/aws-panorama-sample/graph.json`

- Configurazione del pacchetto — `packages/123456789012-SAMPLE_CODE-1.0/package.json`
- Crea script — `3-build-container.sh`

Per aggiornare il codice dell'applicazione

1. Modifica il codice dell'applicazione in`packages/123456789012-SAMPLE_CODE-1.0/src/application.py`.
2. Per creare il contenitore, 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 dalla `assets` cartella e aggiorna la configurazione del pacchetto.

3. Per caricare i pacchetti, esegui. `4-package-application.py`
4. Apri la pagina Applicazioni distribuite della console AWS Panorama.
5. Scegliere un'applicazione.
6. Scegliere Replace (Sostituisci).
7. Completa i passaggi per distribuire l'applicazione. Se necessario, è possibile apportare modifiche al manifesto dell'applicazione, agli stream della videocamera 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 la CLI dell'applicazione AWS Panorama per importarlo come risorsa.

L'esempio seguente utilizza un modello MXNet SSD ResNet 50 che puoi scaricare dal repository di questa guida: GitHub ssd_512_resnet50_v1_voc.tar.gz

Per modificare il modello dell'applicazione di esempio

1. Rinomina la cartella del pacchetto in modo che corrisponda al tuo modello. Ad esempio, a. `packages/123456789012-SSD_512_RESNET50_V1_VOC-1.0/`
2. Aggiorna il nome del pacchetto nelle seguenti posizioni.

   - Manifesto dell'applicazione: `graphs/aws-panorama-sample/graph.json`
   - Configurazione del pacchetto — `packages/123456789012-SSD_512_RESNET50_V1_VOC-1.0/package.json`
3. Nel file di configurazione del pacchetto (`package.json`). Cambia il `assets` valore in una matrice vuota.

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

4. Aprire il file descrittore del pacchetto (`descriptor.json`). Aggiorna i `shape` valori `framework` and in modo che corrispondano al tuo modello.

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

   Il valore della forma 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). `1,3,512,512` I valori e l'ordine dell'array variano tra i modelli.

5. Importa il modello con la CLI dell'applicazione AWS Panorama. La CLI dell'applicazione AWS Panorama copia i file del modello e del descrittore nella `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, esegui. `panorama-cli package-application`

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

7. Aggiorna il codice dell'applicazione. La maggior parte del codice può essere riutilizzata. Il codice specifico per la risposta del modello si trova nel `process_results` metodo.

```
    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)
```

A seconda del modello, potrebbe essere necessario aggiornare anche il `preprocess` metodo.

Preelaborazione delle immagini

Prima di inviare un'immagine al modello, l'applicazione la prepara per l'inferenza ridimensionandola e normalizzando i dati cromatici. 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 primo livello. L'applicazione regola ogni valore di colore convertendolo in un numero compreso tra 0 e 1, sottraendo il valore medio di quel colore e dividendolo per la deviazione standard. Infine, combina i canali di colore e li converte in una NumPy matrice che il modello può elaborare.

Example application.py — Preelaborazione

```
    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 incentrato su 0. Corrisponde alla preelaborazione applicata alle immagini nel set di dati di addestramento, che è un approccio standard ma può variare in base al modello.

Caricamento delle metriche con l'SDK per Python

L'applicazione di esempio utilizza l'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 nel `aws-panorama-sample.yml` CloudFormation modello.

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, `Dockerfile` esegue i comandi per installare le librerie in aggiunta a ciò che viene fornito con l'immagine di base.

Example File Docker

```
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 l' AWS SDK nel codice dell'applicazione, modifica innanzitutto il modello per aggiungere le autorizzazioni per tutte le azioni API utilizzate dall'applicazione. Aggiorna lo CloudFormation stack eseguendolo `1-create-role.sh` ogni 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 questa politica specificando un nome o uno schema per la destinazione `Resource` in un'istruzione separata. Per i dettagli sulle azioni e le risorse supportate da ciascun servizio, consulta Action, resources and condition keys nel Service Authorization Reference

Passaggi successivi

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

- 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 l'archivio di esempi di AWS Panorama.

- github. com/aws-samples/aws-esempi panoramici

Modelli e fotocamere di visione artificiale supportati

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

Per elaborare video e inviare immagini a un modello, AWS Panorama Appliance si connette a un flusso video codificato H.264 con il protocollo RTSP. AWS Panorama verifica la compatibilità di diverse fotocamere comuni.

Sections

Modelli supportati

Quando crei un'applicazione per AWS Panorama, fornisci un modello di apprendimento automatico che l'applicazione utilizza per la visione artificiale. Puoi utilizzare modelli predefiniti e pre-addestrati forniti da framework di modelli, un modello di esempio o un modello creato e addestrato da te.

> **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 AI Neo per compilare il tuo modello di visione artificiale. SageMaker AI Neo è un compilatore che ottimizza i modelli per eseguirli 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 MXNet TensorFlow Apache e quelle supportate per i dispositivi edge da SageMaker AI Neo. Quando crei il tuo modello, puoi utilizzare le versioni del framework elencate nelle note di rilascio di SageMaker AI Neo. In SageMaker AI, puoi utilizzare l'algoritmo di classificazione delle immagini integrato.

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

Telecamere supportate

L'AWS Panorama Appliance supporta flussi video H.264 da telecamere che emettono RTSP su una rete locale. Per flussi di immagini superiori a 2 megapixel, l'appliance ridimensiona l'immagine a 1920x1080 pixel o a una dimensione equivalente che mantenga le proporzioni dello stream.

I seguenti modelli di telecamere sono stati testati per verificarne la compatibilità con AWS Panorama Appliance:

- Asse: M3057-PLVE, M3058-PLVE, P1448-LE, P3225-LV Mk II
- LaView— LV PB3 - 400 W
- Vivotek — 0-H IB936
- Amcrest — IP2 M-841B
- Anoviz — IPC-B850W-3X, IPC-D250W-S
- WGCC — Telecamera PoE da 4 MP ONVIF

Per le specifiche hardware del dispositivo, vedere. Specifiche di AWS Panorama Appliance

Specifiche di AWS Panorama Appliance

L'AWS Panorama Appliance ha le seguenti specifiche hardware. Per altri dispositivi compatibili, consulta la documentazione del produttore.

Componente	Specifiche
Processore e GPU	Nvidia Jetson AGX Xavier con 32 GB di RAM
Ethernet	2 x 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" (197 mm x 243 mm x 40 mm)
Weight	3,7 libbre (1,7 kg)
Alimentazione	100 V-240 V 50-60 Hz AC 65 W
Ingresso di alimentazione	Presa IEC 60320 C6 (3 pin)
Protezione da polvere e liquidi	IP-62
Conformità alle normative EMI/EMC	FCC Part-15 (Stati Uniti)
Limiti termici del tocco	IEC-62368
Temperatura operativa	Da -20°C a 60°C
Umidità operativa	Da 0% a 95% RH
Temperatura di conservazione	Da -20° C a 85° C
Umidità di conservazione	Non controllato per basse temperature. 90% RH ad alta temperatura
Raffreddamento	Estrazione del calore ad aria forzata (ventola)

Componente	Specifiche
Opzioni di montaggio	Montabile su rack o autoportante
Cavo di alimentazione	1,8 metri (6 piedi)
Controllo della potenza	Pulsante
Reimposta	Interruttore momentaneo
Stato e rete LEDs	LED RGB programmabile a 3 colori

Wi-Fi, Bluetooth e memoria su scheda SD sono presenti sull'apparecchio ma non sono utilizzabili.

L'AWS Panorama Appliance include due viti per il montaggio su un rack di server. Puoi montare due appliance side-by-side su un rack da 19 pollici.

Quote del servizio

AWS Panorama applica quote alle risorse che crei nel tuo account e alle applicazioni che distribuisci. Se utilizzi AWS Panorama in più AWS regioni, 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 di applicazioni: 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 la CLI dell'applicazione AWS Panorama o l' AWS SDK con il servizio AWS Panorama, le quote si applicano al numero di chiamate API effettuate. AWS Command Line Interface 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 la console Service Quotas o consulta gli endpoint e le quote di AWS Panorama nel. Riferimenti generali di Amazon Web Services

AWS Panorama autorizzazioni

Puoi utilizzare AWS Identity and Access Management (IAM) per gestire l'accesso al AWS Panorama servizio e alle risorse come appliance e applicazioni. Per gli utenti del tuo account che utilizzano AWS Panorama, gestisci le autorizzazioni in una politica di autorizzazioni che puoi applicare ai ruoli IAM. Per gestire le autorizzazioni per un'applicazione, crei un ruolo e lo assegni all'applicazione.

Per gestire le autorizzazioni per gli utenti del tuo account, utilizza la politica gestita che AWS Panorama fornisce o scrivine una tua. Sono necessarie le autorizzazioni per accedere ad altri AWS servizi per ottenere i registri delle applicazioni e dei dispositivi, visualizzare le metriche e assegnare un ruolo a un'applicazione.

Un' AWS Panorama appliance ha anche un ruolo che le concede l'autorizzazione ad accedere a servizi e risorse. AWS Il ruolo dell'appliance è uno dei ruoli di servizio utilizzati dal servizio per accedere ad altri servizi per conto dell' AWS Panorama utente.

Un ruolo dell'applicazione è un ruolo di servizio separato creato per un'applicazione, per concederle l'autorizzazione a utilizzare AWS i servizi con. AWS SDK per Python (Boto) Per creare un ruolo applicativo, sono necessari i privilegi amministrativi o l'assistenza 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, puoi specificare uno schema per l'Amazon Resource Name (ARN) di un'applicazione che richiede a un utente di includere il proprio nome utente nel nome delle applicazioni che crea. Per le risorse e le condizioni supportate da ciascuna azione, consulta Azioni, risorse e chiavi di condizione AWS Panorama nel Service Authorization Reference.

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

Argomenti

- Policy IAM basate sull'identità per AWS Panorama
- Ruoli del servizio AWS Panorama e risorse multiservizio
- Concessione delle autorizzazioni a un'applicazione

Policy IAM basate sull'identità per AWS Panorama

Per concedere agli utenti del tuo account l'accesso ad AWS Panorama, utilizzi policy basate sull'identità in AWS Identity and Access Management (IAM). Applica policy basate sull'identità ai ruoli IAM associati a un utente. Puoi anche concedere agli utenti di un altro account il permesso di assumere un ruolo nel tuo account e accedere alle tue risorse AWS Panorama.

AWS Panorama fornisce policy gestite che garantiscono l'accesso alle azioni dell'API AWS Panorama 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 necessità, per garantire che gli utenti abbiano accesso alle nuove funzionalità non appena vengono rilasciate.

- AWSPanoramaFullAccess— Fornisce accesso completo ad AWS Panorama, ai punti di accesso AWS Panorama in Amazon S3, alle credenziali delle appliance e ai log delle appliance in Gestione dei segreti AWS Amazon. CloudWatch 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 queste autorizzazioni, aggiungi la seguente policy.

- ResourceGroupsandTagEditorFullAccess— Visualizza la politica

La `AWSPanoramaFullAccess` policy non include l'autorizzazione all'acquisto di dispositivi dalla console AWS Panorama. Per concedere queste autorizzazioni, aggiungi la seguente policy.

- ElementalAppliancesSoftwareFullAccess— Visualizza la politica

Le policy gestite concedono l'autorizzazione alle azioni delle 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. Utilizza la politica di accesso completo come punto di partenza per le tue politiche.

> **Creazione di ruoli di servizio**
>
> La prima volta che utilizzi la console AWS Panorama, hai bisogno dell'autorizzazione per creare il ruolo di servizio utilizzato da AWS Panorama Appliance. Un ruolo di servizio fornisce

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

Per i dettagli sulle risorse e le 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 del servizio AWS Panorama e risorse multiservizio

AWS Panorama utilizza altri servizi AWS per gestire AWS Panorama Appliance, archiviare dati e importare risorse applicative. Un ruolo di servizio fornisce al servizio l'autorizzazione a gestire 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 log e ottenere oggetti dai punti di accesso Amazon S3 creati da AWS Panorama. CloudWatch

  Policy gestita: AWSPanoramaApplianceServiceRolePolicy

Per visualizzare le autorizzazioni associate a ciascun ruolo, utilizza la console 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 al servizio solo l'autorizzazione ad accedere alle AWS IoT risorse `panorama` a suo nome.

Sections

Garantire il ruolo dell'appliance

L'AWS Panorama Appliance utilizza il `AWSPanoramaApplianceServiceRole` ruolo per accedere alle risorse del tuo account. L'appliance è autorizzata a caricare i log su CloudWatch Logs, leggere le credenziali dello streaming della telecamera e accedere agli artefatti dell'applicazione nei punti di accesso Amazon Simple Storage Service (Amazon S3) creati da Gestione dei segreti AWS AWS Panorama.

> **Note**
>
> Le applicazioni non utilizzano le autorizzazioni dell'appliance. Per autorizzare l'applicazione a utilizzare AWS i servizi, crea un ruolo dell'applicazione.

AWS Panorama utilizza lo stesso ruolo di servizio con tutte le appliance del tuo account e non utilizza ruoli su più account. Per un ulteriore livello di sicurezza, puoi modificare la policy di fiducia del ruolo dell'appliance per applicarla in modo esplicito, una buona pratica quando usi i ruoli per concedere a un servizio l'autorizzazione ad accedere alle risorse del tuo account.

Per aggiornare la politica di attendibilità dei ruoli dell'appliance

1. Apri il ruolo dell'appliance nella console IAM: AWSPanoramaApplianceServiceRole
2. Seleziona Modifica relazione di attendibilità.
3. Aggiorna il contenuto della policy, quindi seleziona Update trust policy.

La seguente policy di fiducia include una condizione che garantisce che quando AWS Panorama assume il ruolo di appliance, lo faccia per un'appliance del tuo account. La `aws:SourceAccount` condizione confronta l'ID dell'account specificato da AWS Panorama con quello che includi nella policy.

Example politica di fiducia: account specifico

JSON

```
{
  "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. La `aws:SourceArn` condizione confronta l'ARN dell'appliance specificata da AWS Panorama con quello incluso nella policy.

Example policy di fiducia: appliance singola

JSON

```
{
  "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 l'appliance, 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 sulle migliori pratiche di sicurezza quando i servizi utilizzano i ruoli per accedere alle risorse del tuo account, consulta La problematica del vice confuso nella IAM User Guide.

Uso di altri servizi

AWS Panorama crea o accede a risorse nei seguenti servizi:

- AWS IoT— Cose, policy, certificati e lavori per AWS Panorama Appliance
- Amazon S3: punti di accesso per lo staging di modelli applicativi, codice e configurazioni.
- 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 della IAM User Guide a cui si fa riferimento in questo elenco.

Concessione delle autorizzazioni a un'applicazione

Puoi creare un ruolo per la tua applicazione per concederle l'autorizzazione a chiamare AWS i servizi. Per impostazione predefinita, le applicazioni non dispongono di alcuna autorizzazione. Crei un ruolo applicativo in IAM e lo assegni a un'applicazione durante la distribuzione. Per concedere alla tua applicazione solo le autorizzazioni di cui ha bisogno, crea un ruolo con autorizzazioni per azioni API specifiche.

L'applicazione di esempio include un CloudFormation modello e uno script che creano un ruolo dell'applicazione. È un ruolo di servizio che AWS Panorama può assumere. Questo ruolo concede all'applicazione l'autorizzazione a chiamare per CloudWatch caricare i parametri.

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, consultaAWS Panorama autorizzazioni.

Gestione dell' AWS Panorama appliance

L' AWS Panorama appliance è l'hardware che esegue le applicazioni. La AWS Panorama console viene utilizzata per registrare un dispositivo, aggiornarne il software e distribuirvi applicazioni. Il software dell' AWS Panorama appliance si collega agli stream delle telecamere, invia fotogrammi video all'applicazione e visualizza l'output video su uno schermo collegato.

Dopo aver configurato l'appliance o un altro dispositivo compatibile, si registrano le telecamere per utilizzarle con le applicazioni. È possibile gestire gli stream delle telecamere nella AWS Panorama console. Quando si distribuisce un'applicazione, si scelgono i flussi di telecamere che l'appliance invia all'applicazione per l'elaborazione.

Per i tutorial che introducono l' AWS Panorama appliance con un'applicazione di esempio, consultate. Iniziare con AWS Panorama

Argomenti

Gestione di un'appliance AWS Panorama

Puoi utilizzare la console AWS Panorama per configurare, aggiornare o annullare la registrazione di AWS Panorama Appliance e altri dispositivi compatibili.

Per configurare un'appliance, 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, consultaRegistrazione automatica dei dispositivi.

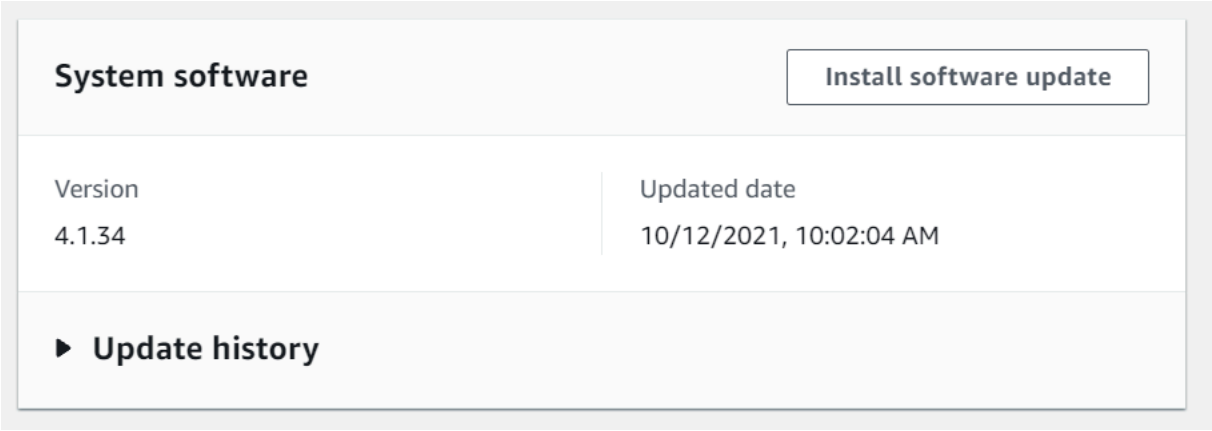
Sections

Aggiorna 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 richiesto, la console richiede di applicarlo. È possibile applicare aggiornamenti opzionali nella pagina Impostazioni dell'appliance.

Per aggiornare il software dell'appliance

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



5. Scegli una nuova versione, quindi scegli Installa.

Annulla la registrazione di un dispositivo

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

Per eliminare un'appliance

1. Apri la pagina Dispositivi della console AWS Panorama.
2. Scegli il nome dell'appliance.
3. Scegli Elimina.
4. Inserisci il nome dell'appliance e scegli Elimina.

Quando elimini un'appliance dal servizio AWS Panorama, i dati sull'appliance non vengono eliminati automaticamente. Un'appliance cancellata non può connettersi ai AWS servizi e non può essere registrata nuovamente finché non viene ripristinata.

Riavviare un dispositivo

È possibile riavviare un dispositivo da remoto.

Per riavviare un dispositivo

1. Apri la pagina Dispositivi della console AWS Panorama.
2. Scegli il nome dell'appliance.
3. Scegliere Reboot (Riavvia).

La console invia un messaggio all'appliance per riavviarlo. Per ricevere il segnale, l'appliance deve essere in grado di connettersi a. AWS IoT Per riavviare un'appliance con l'API AWS Panorama, consulta. Riavviare i dispositivi

Reimposta un'appliance

Per utilizzare un dispositivo in un'altra regione o con un account diverso, è necessario reimpostarlo e rifornirlo 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'apparecchiatura deve essere collegata e spenta. Tieni premuti i pulsanti di accensione e ripristino per cinque secondi. Quando rilasci i pulsanti, la spia di stato lampeggia in arancione. Attendere che la spia di stato lampeggi in verde prima di accendere o scollegare l'apparecchio.

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

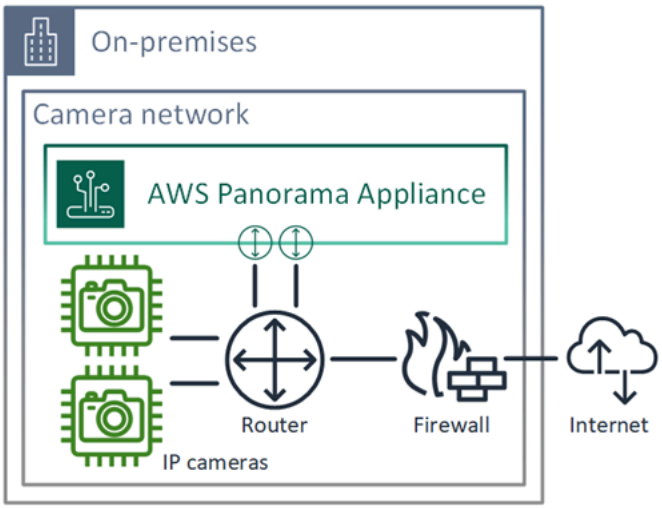
Connessione di AWS Panorama Appliance alla rete

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.

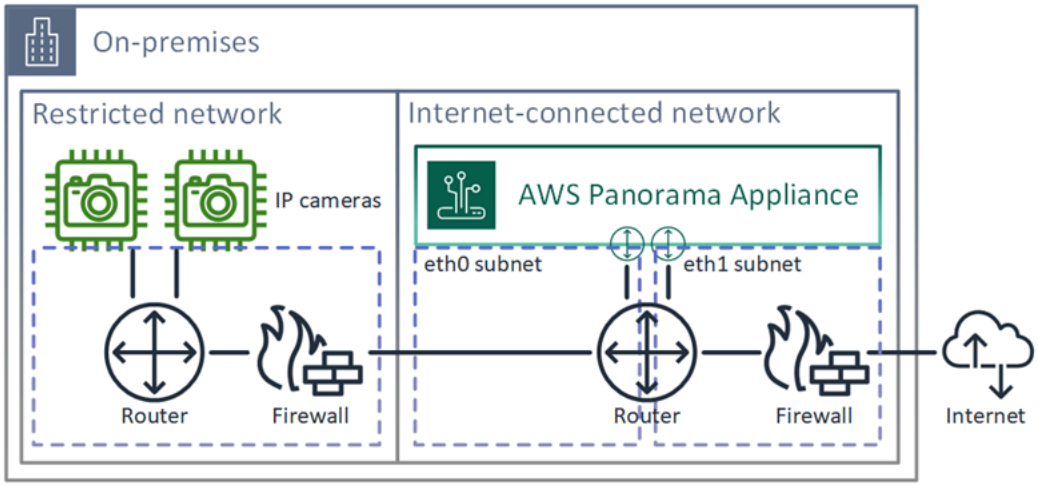


Per informazioni dettagliate sulle porte e sugli endpoint a cui l'appliance deve accedere, vedere e. Configurazione dell'accesso al servizio Configurazione dell'accesso alla rete locale

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 IoT credenziali (HTTPS, porta 443) e sottodomini. `credentials.iot.<region>.amazonaws.com`
- Amazon Elastic Container Registry (HTTPS, porta 443) `dkr.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 usare una Site-to-Site VPN o Direct Connect 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.

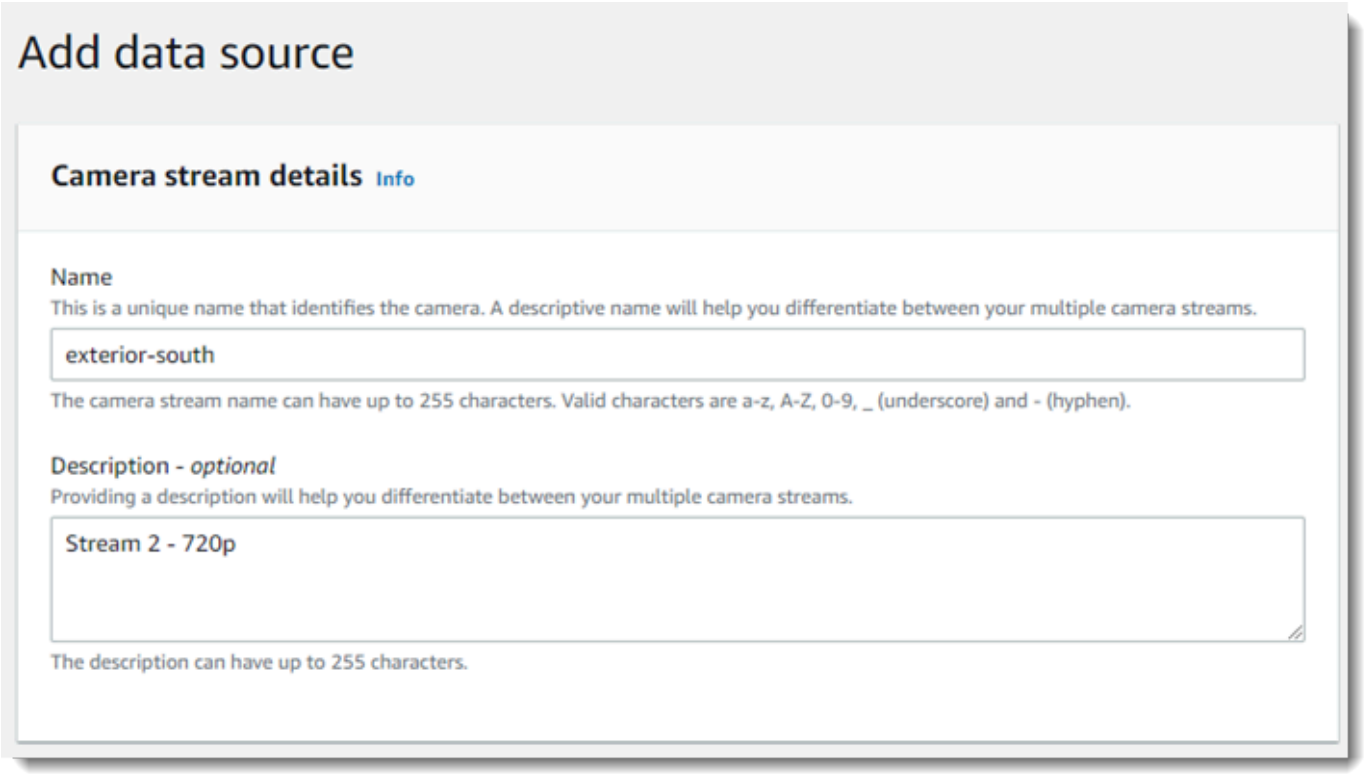
Gestione dei flussi di telecamere in AWS Panorama

Per registrare flussi video come fonti di 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 rete in modo da consentire solo il traffico RTSP a livello locale. Per ulteriori informazioni, consulta Sicurezza in AWS Panorama.

Per registrare lo streaming di una videocamera

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



3. Configura le impostazioni seguenti.

- Nome: un nome per lo stream della telecamera.
- 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, specificate il nome utente e la password.

4. Seleziona Salva.

Per registrare uno stream di telecamere con l'API AWS Panorama, consultaRegistrazione automatica dei dispositivi.

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

Rimuovere uno stream

Puoi eliminare uno stream di telecamere nella console AWS Panorama.

Per rimuovere uno stream di telecamere

1. Apri la pagina Fonti di dati della console AWS Panorama.
2. Scegli uno stream da videocamera.
3. Scegli Elimina fonte di dati.

La rimozione di uno stream di videocamera dal servizio non interrompe l'esecuzione delle applicazioni né elimina le credenziali della videocamera da Secrets Manager. Per eliminare i segreti, usa la console Secrets Manager.

Gestisci le applicazioni su un'appliance AWS Panorama

Un'applicazione è una combinazione di codice, modelli e configurazione. Dalla pagina Dispositivi nella console AWS Panorama, puoi gestire le applicazioni sull'appliance.

Per gestire le applicazioni su un dispositivo AWS Panorama

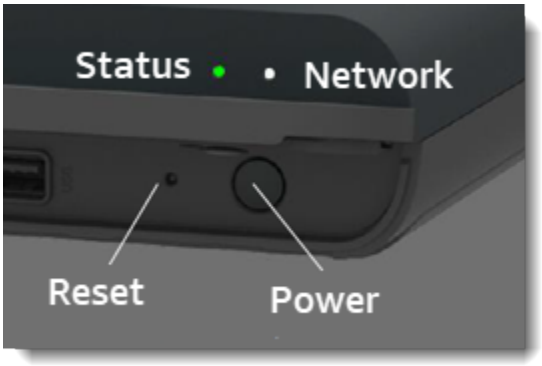
1. Apri la pagina Dispositivi della console AWS Panorama.
2. Scegli un'appliance.

La pagina Applicazioni distribuite 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 Appliance

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



Indicatore luminoso di stato

LEDs Cambia colore e lampeggia per indicare lo stato. Un lampeggiamento lento si verifica una volta ogni tre secondi. Un lampeggiamento rapido avviene una volta al secondo.

Stati del LED di stato

- Verde lampeggiante rapidamente: l'appliance si sta avviando.
- Verde fisso: l'apparecchiatura funziona normalmente.
- Blu che lampeggia lentamente: l'appliance sta copiando i file di configurazione e sta tentando di registrarsi con. AWS IoT
- Blu lampeggiante rapidamente: l'appliance sta copiando un'immagine di registro su un'unità USB.
- Rosso lampeggiante rapido: l'apparecchio ha riscontrato un errore durante l'avvio o si è surriscaldato.
- Arancione che lampeggia lentamente: l'appliance sta ripristinando l'ultima versione del software.
- Arancione lampeggiante rapidamente: l'appliance sta ripristinando la versione minima del software.

Luce di rete

Il LED di rete ha i seguenti stati:

Stati dei LED di rete

- Verde fisso: è collegato un cavo Ethernet.

- Verde lampeggiante: l'appliance sta comunicando tramite la rete.
- Rosso fisso: non è collegato un cavo Ethernet.

Pulsanti di accensione e ripristino

I pulsanti di accensione e ripristino si trovano sulla 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 resettare un apparecchio

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

   - Ripristino completo: premi a lungo l'accensione e ripristina.

     Ripristina la versione minima del software ed elimina tutti i file di configurazione e le applicazioni.
   - Ripristina la versione più recente del software: premi brevemente il pulsante di ripristino.

     Riapplica l'ultimo aggiornamento software all'appliance.
   - Ripristina la versione minima del software: premere a lungo su reset.

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

Il ripristino di un'appliance non la elimina dal servizio AWS Panorama. Per ulteriori informazioni, consulta Annulla la registrazione di un dispositivo.

Gestione delle AWS Panorama applicazioni

Le applicazioni vengono eseguite sull' AWS Panorama appliance per eseguire attività di visione artificiale su flussi video. È possibile creare applicazioni di visione artificiale combinando codice Python e modelli di machine learning e distribuirle nell' AWS Panorama Appliance tramite 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, usi la CLI dell'applicazione AWS Panorama, la importi nel tuo account, crei il contenitore, carichi e registri le risorse e crei un'istanza dell'applicazione. Questo argomento illustra in dettaglio ciascuno di questi passaggi e descrive ciò che accade in background.

Se non hai ancora distribuito un'applicazione, consulta Iniziare con AWS Panorama la procedura dettagliata.

Per ulteriori informazioni sulla personalizzazione e l'estensione dell'applicazione di esempio, consulta. AWS Panorama Applicazioni edili

Sections

Installa la CLI dell'applicazione AWS Panorama

Per installare la CLI dell'applicazione AWS Panorama e utilizzare AWS CLI pip.

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

Per creare immagini di applicazioni con la CLI dell'applicazione AWS Panorama, hai bisogno di Docker. Su Linux sono necessarie anche `qemu` le relative librerie di sistema. Per ulteriori informazioni sull'installazione e la configurazione della CLI 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, vedi. WSL2 Configurazione di un ambiente di sviluppo in Windows

Importazione di un'applicazione

Se stai lavorando con un'applicazione di esempio o un'applicazione fornita da terze parti, utilizza la CLI dell'applicazione AWS Panorama per importare l'applicazione.

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

Questo comando rinomina i pacchetti di applicazioni con l'ID dell'account. I nomi dei pacchetti iniziano con l'ID account dell'account su cui vengono distribuiti. Quando si distribuisce un'applicazione su più account, è necessario importare e impacchettare l'applicazione separatamente per ogni account.

Ad esempio, l'applicazione di esempio di questa guida è 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 dell'account che la CLI deduce dalle credenziali dell'area 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 nel manifesto dell'applicazione (`graph.json`), che fa riferimento a essi. Puoi confermare l'ID del tuo account chiamando `aws sts get-caller-identity` con AWS 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 del contenitore Docker, che include il codice dell'applicazione e le librerie installate nel Dockerfile. Usa il `build-container` comando CLI dell'applicazione AWS Panorama per creare un'immagine Docker ed esportare un'immagine del filesystem.

```
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 filesystem in un archivio nella cartella. `.tar.gz` `assets` La CLI 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. La CLI dell'applicazione AWS Panorama aggiunge anche un blocco alla configurazione del pacchetto che registra i nomi dei due asset. Questi nomi vengono utilizzati dall'appliance durante il processo di distribuzione.

Example Packages/123456789012-sample_code-1.0/Package.json — con asset block

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

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

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

Importa un modello

L'applicazione potrebbe avere un archivio dei modelli nella cartella delle risorse o scaricarlo separatamente. Se avete un nuovo modello, un modello aggiornato o un file descrittore del modello aggiornato, utilizzate 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 avete solo bisogno di aggiornare il file descrittore, potete 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, lo script seguente mostra come eseguire questa operazione con l'app di esempio.

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

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

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

Caricate le risorse dell'applicazione

Per caricare e registrare le risorse dell'applicazione, che includono l'archivio del modello, l'archivio del filesystem del 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 asset 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
```

La CLI 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 al riguardo tramite l'DescribePackageAPI. La CLI 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.

Distribuisci 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 telecamere passare al codice dell'applicazione e configuri le opzioni fornite dallo sviluppatore dell'applicazione.

Per distribuire un'applicazione

1. Apri la pagina Applicazioni distribuite della console AWS Panorama.
2. Scegli Deploy application.
3. Incolla il contenuto del manifesto dell'applicazione nell'editor di testo. `graph.json` Scegli Next (Successivo).
4. Immettete un nome e una descrizione.
5. Scegli Procedi alla distribuzione.
6. Scegli Inizia la distribuzione.
7. Se l'applicazione utilizza un ruolo, selezionalo dal menu a discesa. Scegli Next (Successivo).
8. Scegli Seleziona dispositivo, quindi scegli il tuo dispositivo. Scegli Next (Successivo).

9. Nel passaggio Seleziona fonti di dati, scegli Visualizza input e aggiungi lo stream della videocamera come fonte di dati. Scegli Next (Successivo).
10. Nella fase Configura, configura tutte le impostazioni specifiche dell'applicazione definite dallo sviluppatore. Scegli Next (Successivo).
11. Scegli Distribuisci, quindi scegli Fine.
12. Nell'elenco delle applicazioni distribuite, scegli l'applicazione per monitorarne lo stato.

Il processo di distribuzione richiede 15-20 minuti. L'output dell'appliance può rimanere vuoto per un periodo prolungato durante l'avvio dell'applicazione. Se si verifica un errore, consultaRisoluzione dei problemi.

Automatizza la distribuzione delle applicazioni

È possibile 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 override che specifica le sostituzioni in fase di implementazione dei valori nel manifesto dell'applicazione. L'utilizzo di un file overrides consente di utilizzare lo stesso manifesto dell'applicazione per distribuire l'applicazione con flussi di telecamere diversi e configurare altre impostazioni specifiche dell'applicazione.

Per ulteriori informazioni e per esempi di script per ciascuno dei passaggi descritti in questo argomento, vedere. Automatizza la distribuzione delle applicazioni

Gestione delle applicazioni nella console AWS Panorama

Usa la console AWS Panorama per gestire le applicazioni distribuite.

Sections

Aggiorna o copia un'applicazione

Per aggiornare un'applicazione, utilizzate l'opzione Sostituisci. Quando sostituite un'applicazione, potete aggiornarne il codice o i modelli.

Per aggiornare un'applicazione

1. Apri la pagina Applicazioni distribuite della console AWS Panorama.
2. Scegliere un'applicazione.
3. Scegliere Replace (Sostituisci).
4. Segui le istruzioni per creare una nuova versione o applicazione.

C'è anche un'opzione Clone che funziona in modo simile a Replace, 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 ripulire le versioni inutilizzate delle applicazioni, eliminale dai dispositivi.

Eliminazione di un'applicazione

1. Apri la pagina Applicazioni distribuite della console AWS Panorama.
2. Scegliere un'applicazione.
3. Scegli Elimina dal dispositivo.

Configurazione del pacchetto

Quando utilizzi il comando CLI dell'applicazione AWS Panoramapanorama-cli package-application, l'interfaccia a riga di comando carica gli asset dell'applicazione su Amazon S3 e li registra con AWS Panorama. Le risorse includono file binari (immagini e modelli di container) e file descrittori, che AWS Panorama Appliance scarica durante la distribuzione. Per registrare le risorse di un pacchetto, fornisci un file di configurazione del pacchetto separato che definisce il pacchetto, i suoi asset e la sua interfaccia.

L'esempio seguente mostra una configurazione di pacchetto per un nodo di codice con un input e un output. L'ingresso video fornisce l'accesso ai dati delle immagini provenienti dallo stream di una telecamera. Il nodo di uscita invia le 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 la CLI dell'applicazione AWS Panorama ha caricato su Amazon S3. Se importi 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 tuo account. Durante l'esecuzione`panorama-cli package-application`, la CLI dell'applicazione AWS Panorama compila questa sezione con i valori corretti.

Il manifesto dell'applicazione AWS Panorama

Quando distribuisci un'applicazione, fornisci un file di configurazione chiamato manifesto dell'applicazione. Questo file definisce l'applicazione come un grafico con nodi e bordi. Il manifesto dell'applicazione fa parte del codice sorgente dell'applicazione ed è memorizzato nella `graphs` directory.

Example graphs/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 bordi, che specificano le mappature tra gli input e gli output dei nodi. L'uscita di un nodo si collega all'ingresso di un altro, formando un grafico.

Schema JSON

Il formato del manifesto dell'applicazione e dei documenti di override 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 nell'archivio di questa guida. GitHub

- Schema JSON — /resources aws-panorama-developer-guide

Nodi applicativi

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 integrato.

Nel seguente esempio, `code_node` `model_node` fai riferimento al codice di esempio e ai pacchetti di modelli inclusi nell'applicazione di esempio. `camera_node`utilizza un pacchetto fornito da AWS Panorama per creare un segnaposto per uno stream 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 all'ingresso di un altro. Nell'esempio seguente, il primo bordo mappa l'output da un nodo di flusso della telecamera all'ingresso di un nodo di codice dell'applicazione. I nomi `video_in` e `video_out` sono definiti 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 dell'applicazione, si utilizzano gli `outputs` attributi `inputs` and per ottenere immagini dal flusso di input e inviarle 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 si riferisce a un pacchetto definito da AWS Panorama, che puoi usare come segnaposto nel manifesto dell'applicazione. AWS Panorama offre due tipi di nodi astratti.

- Flusso della telecamera: scegli lo stream della videocamera utilizzato dall'applicazione durante la distribuzione.

  Nome del pacchetto — `panorama::abstract_rtsp_media_source`

  Nome dell'interfaccia — `rtsp_v1_interface`

- Uscita HDMI: indica che l'applicazione emette video.

  Nome del pacchetto — `panorama::hdmi_data_sink`

  Nome dell'interfaccia — `hdmi0`

L'esempio seguente mostra un set di base di pacchetti, nodi e edge per un'applicazione che elabora i flussi delle telecamere e trasmette i video su uno schermo. Il nodo telecamera, che utilizza l'interfaccia del `abstract_rtsp_media_source` pacchetto in AWS Panorama, può accettare più flussi di telecamere come input. Il nodo di uscita, che fa riferimento`hdmi_data_sink`, consente al codice dell'applicazione di accedere a un buffer video 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"
            }
        ]
    }
}
```

Parametri dell'applicazione

I parametri sono nodi di tipo base che possono essere sostituiti durante la distribuzione. Un parametro può avere un valore predefinito e un decoratore, che indica all'utente dell'applicazione come configurarlo.

Tipi di parametro

- `string`— Una stringa. Ad esempio `DEBUG`.
- `int32`— Un numero intero. Ad esempio, `20`.
- `float32`— Un numero in virgola mobile. Ad esempio, `47.5`.
- `boolean`— `true` oppure. `false`

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 in fase di implementazione con sostituzioni. Per ulteriori informazioni, consulta Configurazione del tempo di implementazione con sostituzioni.

Configurazione del tempo di implementazione con sostituzioni

I parametri e i nodi astratti vengono configurati durante la distribuzione. Se utilizzi la console AWS Panorama per la distribuzione, puoi specificare un valore per ogni parametro e scegliere uno stream di telecamere come input. Se utilizzi l'API AWS Panorama per distribuire applicazioni, specifichi queste impostazioni con un documento di override.

Un documento di override ha una struttura simile a quella di un manifesto dell'applicazione. Per i parametri con tipi di base, si definisce un nodo. Per i flussi di telecamere, definite un nodo e un pacchetto che mappano a un flusso di telecamere registrato. Quindi si definisce un'eccezione per ogni nodo che specifica il nodo del manifesto dell'applicazione che sostituisce.

Example sovrascrive il file.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 fotocamera astratto. `node0verrides`Indica ad AWS Panorama quali nodi in questo documento hanno la precedenza su quali nel manifesto dell'applicazione.

AWS Panorama Applicazioni edili

Le applicazioni vengono eseguite sull' AWS Panorama appliance per eseguire attività di visione artificiale su flussi video. È possibile creare applicazioni di visione artificiale combinando codice Python e modelli di machine learning e distribuirle nell' AWS Panorama Appliance tramite Internet. Le applicazioni possono inviare video a un display o utilizzare l'SDK AWS per inviare risultati ai servizi AWS.

Un modello analizza 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 di indovinare. Puoi addestrare i modelli con i tuoi dati di immagine o iniziare con un campione.

Il codice dell'applicazione elabora le immagini fisse provenienti dallo stream di una telecamera, le invia a un modello ed elabora il risultato. Un modello può 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 i risultati a un AWS servizio per l'archiviazione o l'ulteriore elaborazione.

Per ottenere immagini da uno stream, interagire con un modello e generare video, il codice dell'applicazione utilizza l' AWS Panorama Application SDK. L'SDK dell'applicazione è una libreria Python che supporta modelli generati PyTorch con, MXNet Apache e. TensorFlow

Argomenti

- Modelli di visione artificiale
- Creazione di un'immagine dell'applicazione
- Chiamare i servizi AWS dal codice dell'applicazione
- L'SDK per applicazioni AWS Panorama
- Esecuzione di più thread
- Servire il 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 prima le immagini di tali oggetti attraverso l'addestramento. Un modello di visione artificiale utilizza un'immagine come input e restituisce informazioni sugli oggetti rilevati, ad esempio 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.

Sections

Utilizzo di modelli nel codice

Un modello restituisce uno o più risultati, che possono includere probabilità per le classi rilevate, informazioni sulla posizione e altri dati. L'esempio seguente mostra come eseguire l'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 matrice 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.

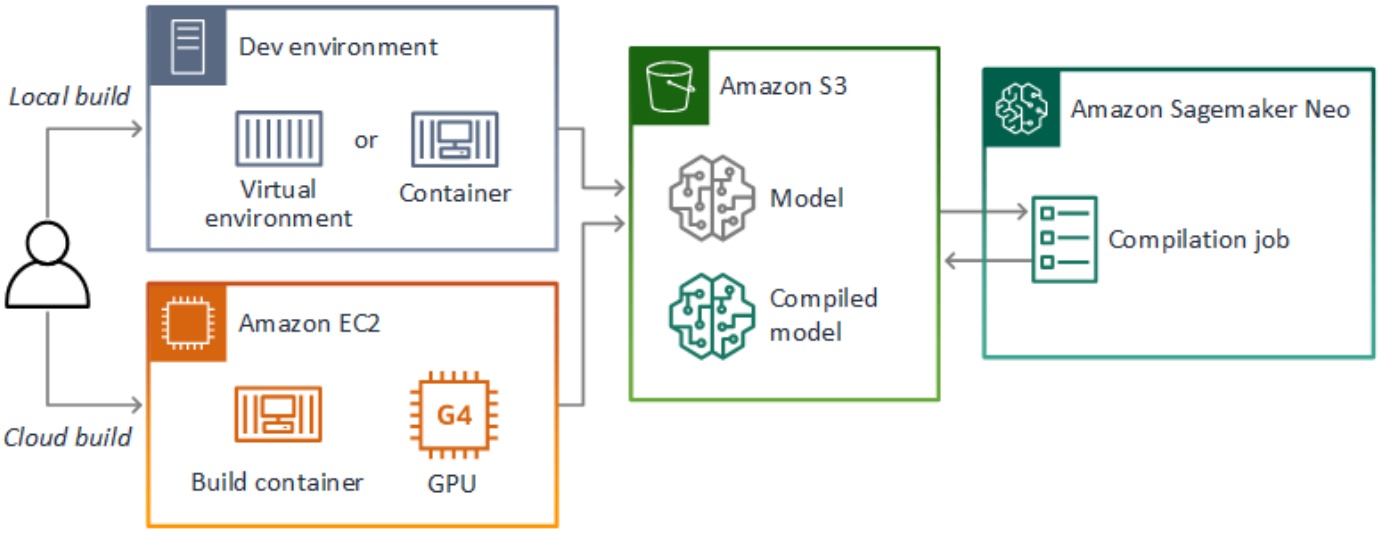
Creazione di un modello personalizzato

Puoi usare modelli che crei in PyTorch MXNet Apache e TensorFlow nelle applicazioni AWS Panorama. In alternativa alla creazione e alla formazione di modelli nell' SageMaker intelligenza artificiale, 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 informazioni dettagliate sulle versioni del framework e sui formati di file supportati da SageMaker AI Neo, consulta Supported Frameworks nella Amazon SageMaker AI Developer Guide.

L'archivio di questa guida fornisce un'applicazione di esempio che illustra 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 creare il modello su un' EC2istanza Amazon.

- Modello di applicazione di esempio personalizzato



AWS Panorama utilizza SageMaker AI Neo per compilare modelli da utilizzare su AWS Panorama Appliance. Per ogni framework, usa il formato supportato da SageMaker AI Neo e raccogli il modello in un `.tar.gz` archivio.

Per ulteriori informazioni, consulta Compilare e distribuire modelli con Neo nella Amazon SageMaker AI 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 applicative, la configurazione del pacchetto indica al servizio AWS Panorama dove il modello e il descrittore sono archiviati 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**
>
> Specificate solo la versione principale e secondaria della versione del framework. Per un elenco delle versioni supportate PyTorch, di Apache MXNet e delle TensorFlow versioni, consulta Framework supportati.

Per importare un modello, usa il comando CLI `import-raw-model` dell'applicazione AWS Panorama. Se apporti modifiche al modello o al suo descrittore, devi eseguire nuovamente questo comando per aggiornare gli asset dell'applicazione. Per ulteriori informazioni, consulta Modifica del modello di visione artificiale.

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

Addestramento dei modelli

Quando addestrate un modello, utilizzate immagini provenienti dall'ambiente di destinazione o da un ambiente di test molto simile a quello 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 retroilluminazione.
- 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 di un soggetto sono visibili. Ad esempio, una fotocamera a soffitto con obiettivo grandangolare mostrerà la parte superiore di un soggetto quando è al centro dell'inquadratura e una visione distorta del lato del soggetto man mano che si allontana dal centro.

Per risolvere questi problemi, è possibile 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 del mondo reale. Se un modello deve funzionare in situazioni di illuminazione e con una varietà di telecamere, sono necessari più dati per l'addestramento. Oltre a raccogliere più immagini, è possibile ottenere più dati di addestramento creando varianti delle immagini esistenti che sono distorte o hanno un'illuminazione diversa.

Creazione di un'immagine dell'applicazione

L'AWS Panorama Appliance esegue le applicazioni come file system container esportati da un'immagine creata da te. Specifichi le dipendenze e le risorse dell'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 la CLI dell'applicazione AWS Panorama. Il seguente esempio tratto dall'applicazione di esempio di questa guida illustra 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 del Dockerfile.

- `FROM`— Carica l'immagine di base dell'applicazione ()`public.ecr.aws/panorama/panorama-application`.
- `WORKDIR`— Imposta la directory 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`.

È possibile utilizzare altre istruzioni, come `ADD` e`ARG`, 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 dall'immagine. Utilizza l'immagine solo per esportare un filesystem, che viene trasferito all'appliance.

Specifica delle dipendenze

`requirements.txt`è un file di requisiti Python che specifica le librerie utilizzate dall'applicazione. L'applicazione di esempio utilizza Open CV e. AWS SDK per 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 in`/usr/local/lib`, in modo che possano essere importate dal codice dell'applicazione.

Storage locale

AWS Panorama riserva la `/opt/aws/panorama/storage` directory per lo storage delle applicazioni. L'applicazione può creare e modificare file su questo percorso. I file creati nella directory di archiviazione persistono anche dopo i riavvii. Le altre posizioni dei file temporanei 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, la CLI viene eseguita `docker build` nella directory del pacchetto. Questo crea un'immagine dell'applicazione che contiene il codice dell'applicazione. La CLI crea quindi un contenitore, esporta il relativo filesystem, 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 asset che la CLI 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 relativo checksum SHA-256. Questo nome viene utilizzato come identificatore univoco per l'asset quando è archiviato in Amazon S3.

Chiamare i servizi AWS dal codice dell'applicazione

Puoi utilizzarli AWS SDK per Python (Boto) per richiamare i servizi AWS dal codice dell'applicazione. Ad esempio, se il tuo modello rileva qualcosa di insolito, puoi pubblicare parametri su Amazon, inviare una notifica con Amazon SNS CloudWatch, 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.

Per impostazione predefinita, l'appliance non dispone dell'autorizzazione per accedere a nessun servizio AWS. Per concederle l'autorizzazione, crea un ruolo per l'applicazione e assegnalo all'istanza dell'applicazione durante la distribuzione.

Sections

Uso di Amazon S3

Puoi usare Amazon S3 per archiviare i risultati di 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'argomento AWS IoT MQTT

È possibile utilizzare l'SDK for Python (Boto3) per inviare messaggi a un argomento MQTT in. AWS IoTNell'esempio seguente, l'applicazione invia un messaggio a un argomento che prende il nome dal nome dell'oggetto dell'appliance, che è possibile trovare nella console.AWS IoT

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

Scegliete un nome che indichi l'ID del dispositivo o un altro identificatore a vostra scelta.
Per pubblicare messaggi, l'applicazione necessita dell'autorizzazione per effettuare chiamate`iot:Publish`.

Per monitorare una coda MQTT

1. Apri la pagina di test AWS IoT della console.
2. Per l'argomento Abbonamento, inserisci il nome dell'argomento. Ad esempio `panorama/panorama_my-appliance_Thing_a01e373b`.
3. Scegli Subscribe to topic (Effettua sottoscrizione all'argomento).

L'SDK per applicazioni AWS Panorama

L'SDK per applicazioni AWS Panorama è una libreria Python per lo sviluppo di applicazioni AWS Panorama. Nel codice dell'applicazione, usi l'SDK dell'applicazione AWS Panorama per caricare un modello di visione artificiale, eseguire inferenze e inviare video a un monitor.

> **Note**
>
> Per assicurarti di avere accesso alle funzionalità più recenti dell'SDK per applicazioni AWS Panorama, aggiorna il software dell'appliance.

Per i dettagli sulle classi definite dall'SDK dell'applicazione e sui relativi metodi, consulta Application SDK reference.

Sections

- Aggiungere testo e riquadri al video in uscita

Aggiungere testo e riquadri al video in uscita

Con l'SDK AWS Panorama, puoi inviare un flusso video a un display. Il video può includere testo e riquadri che mostrano l'output del modello, lo stato corrente dell'applicazione o altri dati.

Ogni oggetto dell'`video_in`array è un'immagine proveniente da un flusso di telecamere collegato all'appliance. Il tipo di oggetto è`panoramasdk.media`. Dispone di metodi per aggiungere testo e riquadri rettangolari all'immagine, che è quindi possibile assegnare all'`video_out`array.

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

```
        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 un riquadro all'immagine di output, usa`add_rect`. Questo metodo richiede 4 valori compresi tra 0 e 1, che indicano la posizione degli angoli superiore sinistro e inferiore destro del riquadro.

```
        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, è possibile creare un thread che serva il traffico HTTP per il debug o un thread che monitora i risultati dell'inferenza e invia i dati a. AWS

Per eseguire più thread, si utilizza il modulo threading della 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 applicativo e lo utilizza per eseguire tre thread.

Example Packages/123456789012-debug_server-1.0/application.py — Ciclo 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.')
```

Quando tutti i thread vengono chiusi, l'applicazione si riavvia automaticamente. Il `run_cv` loop elabora le immagini provenienti dai flussi della telecamera. Se riceve un segnale di arresto, interrompe il processo di debugger, che esegue un server HTTP e non può spegnersi da solo. Ogni

thread deve gestire i propri errori. Se un errore non viene rilevato e registrato, il thread si chiude silenziosamente.

Example Packages/123456789012-debug_server-1.0/application.py — Ciclo 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'oggetto dell'applicazione. `self` Per riavviare il ciclo di elaborazione dell'applicazione, il thread del debugger chiama il metodo. `stop` Questo metodo imposta un `terminate` attributo che segnala agli altri thread di chiudersi.

Example Packages/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)
```

Servire il traffico in entrata

Puoi monitorare o eseguire il debug delle applicazioni localmente eseguendo un server HTTP insieme al codice dell'applicazione. Per servire il traffico esterno, mappi 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 delle porte sull'appliance comporta un rischio implicito per la sicurezza. Quando si utilizza questa funzionalità, è necessario adottare misure aggiuntive per proteggere l'appliance dal traffico esterno e proteggere le comunicazioni tra i client autorizzati e l'appliance.
> 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'appliance. Queste porte, una volta aperte, possono ricevere traffico da qualsiasi client routabile. Quando si distribuisce l'applicazione, si specificano le porte da aprire e si associano le porte dell'appliance alle porte del contenitore dell'applicazione. Il software dell'appliance inoltra il traffico al contenitore e invia le risposte al richiedente. Le richieste vengono ricevute sulla porta dell'appliance specificata e le risposte vengono inviate su una porta temporanea casuale.

Configurazione delle porte in entrata

Le mappature delle porte vengono specificate in tre punti della configurazione dell'applicazione. Per quanto riguarda il pacchetto di codice`package.json`, si specifica la porta su cui il nodo di codice ascolta in un blocco. `network` L'esempio seguente dichiara che il nodo è in ascolto sulla porta 80.

Example Pacchetti/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'appliance a una porta del contenitore di codice dell'applicazione. L'esempio seguente aggiunge una regola che mappa la porta 8080 sul dispositivo alla porta 80 del contenitore. `code_node`

Example graphs/my-app/graph.json

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

Quando distribuisci l'applicazione, specifichi le stesse regole nella console AWS Panorama o con un documento di override passato all'CreateApplicationInstanceAPI. È necessario fornire questa configurazione al momento della distribuzione per confermare che si desidera aprire le porte sull'appliance.

Example graphs/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.

Servire il traffico

Con le porte aperte sul contenitore, puoi aprire un socket o eseguire un server per gestire le richieste in arrivo. L'`debug-server`esempio mostra un'implementazione di base di un server HTTP in esecuzione insieme al codice di un'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 Packages/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 le richieste GET lungo il percorso per recuperare alcune informazioni sull'applicazione. `/status` Accetta anche una richiesta POST per `/restart` riavviare l'applicazione.

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

Example Packages/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 quando escono.

Example Packages/123456789012-debug_server-1.0/application.py — Ciclo 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, consulta le istruzioni nell'archivio di questa guida. GitHub

Utilizzo della GPU

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

> **Important**
>
> Se abiliti l'accesso tramite GPU, non puoi eseguire nodi modello in nessuna applicazione sull'appliance. Per motivi di sicurezza, l'accesso alla GPU è limitato quando l'appliance esegue un modello compilato con AI 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 averlo creato con la CLI dell'applicazione AWS Panorama. L'esempio seguente mostra il `requirements` blocco che aggiunge l'accesso tramite GPU al nodo del codice dell'applicazione.

Example package.json con blocco requirements

```
{
    "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 pacchettizzazione nel flusso di lavoro di sviluppo.

Per distribuire 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 la configurazione della risorsa del contenitore e del pacchetto, usa il `package-application` comando.

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

4. Distribuire l'applicazione.

Per esempi di applicazioni che utilizzano l'accesso tramite GPU, visita il aws-panorama-samples GitHub repository.

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, puoi configurare un ambiente di sviluppo utilizzando Docker Desktop con Windows Subsystem per Linux e Ubuntu. Questo tutorial illustra il processo di configurazione di un ambiente di sviluppo che è stato testato con strumenti AWS Panorama e applicazioni di esempio.

Sections

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 successive) o Windows 11
- Funzionalità di Windows
  - Sottosistema Windows per Linux
  - Hyper-V
  - Piattaforma di macchina virtuale

Questo tutorial è stato sviluppato con le seguenti versioni del software.

- Ubuntu 20.04
- Python 3.8.5
- Docker 20.10.8

Installa WSL 2 e Ubuntu

Se disponi di Windows 10 versione 2004 e successive (build 19041 e successive), puoi installare WSL 2 e Ubuntu 20.04 con il seguente comando. PowerShell

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

Per le versioni precedenti di Windows, segui le istruzioni nella documentazione di WSL 2: Procedura di installazione manuale per versioni precedenti

Installa Docker

Per installare Docker Desktop, scarica ed esegui il pacchetto di installazione da hub.docker.com. Se riscontri problemi, segui le istruzioni sul sito Web di Docker: Docker Desktop WSL 2 backend.

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 hai installato altre distribuzioni Linux prima di eseguire questo tutorial, abilita Docker nella distribuzione Ubuntu appena installata nel menu delle impostazioni di Docker Desktop in Risorse, integrazione WSL.

Configura Ubuntu

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

Per completare la configurazione dell'ambiente di sviluppo, aggiorna il software della macchina virtuale e installa 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 le librerie Python con pip.

```
$ pip3 install awscli panoramacli
```

4. Apri un nuovo terminale, quindi esegui `aws configure` per configurare. AWS CLI

```
$ aws configure
```

   Se non disponi delle chiavi di accesso, puoi generarle nella console IAM.

Infine, scarica e importa l'applicazione di esempio.

Per scaricare l'applicazione di esempio

1. Scaricate ed estraete 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 di applicazioni e caricare pacchetti su AWS Panorama.

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

L'interfaccia a riga di comando dell'applicazione AWS Panorama carica i pacchetti e li registra con il servizio AWS Panorama. Ora puoi distribuire l'app di esempio con la console AWS Panorama.

Passaggi successivi

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

Per accedere al file system della macchina virtuale, apri File explorer ed entra `\\wsl$` nella barra di navigazione. Questa directory contiene un collegamento al file system (`Ubuntu-20.04`) e ai file system della macchina virtuale per i dati di Docker. Sotto`Ubuntu-20.04`, la tua directory utente si trova in`home\`*`username`*.

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

Con Visual Studio Code, puoi modificare il codice dell'applicazione nell'ambiente di sviluppo ed eseguire comandi con un terminale integrato. Per installare Visual Studio Code, visita code.visualstudio.com. Dopo l'installazione, aggiungi l'estensione Remote WSL.

Il terminale Windows è un'alternativa al terminale Ubuntu standard in cui hai eseguito i comandi. Supporta più schede e può essere eseguito PowerShell, Command Prompt e terminali per qualsiasi altra varietà di Linux installata. Supporta il copia e incollaCtrl+V, la Ctrl+C funzionalità cliccabile URLs e altri miglioramenti utili. Per installare Windows Terminal, visita microsoft.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 AWS Command Line Interface o l' AWS SDK, puoi sviluppare script o applicazioni che gestiscono risorse e distribuzioni. L' GitHub archivio di questa guida include script che puoi utilizzare come punto di partenza per il tuo codice.

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

Sections

Registrazione automatica dei dispositivi

Per effettuare il provisioning di un'appliance, utilizza l'ProvisionDeviceAPI. La risposta include un file ZIP con la configurazione del dispositivo e le credenziali temporanee. Decodifica il file e salvalo in un archivio con il prefisso. `certificates-omni_`

Example provision-device.sh

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

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, utilizzate l'CreateNodeFromTemplateJobAPI. Questa API utilizza 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 in 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, puoi 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 delle appliance con l'API AWS Panorama.

Visualizza i dispositivi

Per ottenere un elenco di dispositivi con dispositivo IDs, 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
}
```

Aggiorna il software dell'appliance

Se la `LatestSoftware` versione è più recente della`CurrentSoftware`, è possibile aggiornare il dispositivo. Utilizza 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. Monitora lo stato di avanzamento 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 lavori in esecuzione, usa 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, consulta check-updates.sh nell' GitHub archivio di questa guida.

Riavviare i dispositivi

Per riavviare un dispositivo, utilizzare l'API. CreateJobForDevices

```
$ 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 per il riavvio 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, usi sia la AWS Command Line Interface CLI dell'applicazione AWS Panorama che. Dopo aver creato il contenitore dell'applicazione, lo carichi insieme ad altre risorse su un punto di accesso Amazon S3. Quindi distribuisci l'applicazione con l'CreateApplicationInstanceAPI.

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

Sections

Costruisci il contenitore

Per creare il contenitore dell'applicazione, usa il `build-container` comando. Questo comando crea un contenitore Docker e lo salva come file system compresso nella cartella. `assets`

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 dalla riga di comando per inserire l'argomento del percorso digitando una parte del percorso e quindi premendo. TAB

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

Caricate il contenitore e registrate i nodi

Per caricare l'applicazione, usa il `package-application` comando. Questo comando carica le risorse dalla `assets` cartella su 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 gli asset container e descrittori a cui fa riferimento la configurazione del pacchetto (`package.json`) in ogni pacchetto e registra i pacchetti come nodi in AWS Panorama. Fai quindi riferimento a questi nodi nel manifesto dell'applicazione (`graph.json`) per distribuire l'applicazione.

Distribuzione dell'applicazione

Per distribuire l'applicazione, si utilizza l'CreateApplicationInstanceAPI. 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 una risorsa statica nell'origine dell'applicazione, dove il manifesto di sostituzione fornisce impostazioni relative alla fase di distribuzione che personalizzano la distribuzione.
- `DefaultRuntimeContextDevice`— Il dispositivo di destinazione.
- `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. A tale scopo, lo script carica i manifest da un file come stringa e utilizza lo strumento jq per costruire il documento annidato.

Example 5-deploy.sh — compone i 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 utilizza l'ListDevicesAPI 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 da. AWS CloudFormation Se l'applicazione è già distribuita sul dispositivo, lo script ottiene l'ID di quell'istanza dell'applicazione da un file locale.

Example 5-deploy.sh — ARN del ruolo e argomenti di sostituzione

```
# 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: distribuisce 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, utilizza l'ListApplicationInstancesAPI. 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
```

Una volta completata la distribuzione, puoi visualizzare i log chiamando l'API Amazon CloudWatch Logs. Lo script di visualizzazione dei log utilizza l'API Logs. CloudWatch `GetLogEvents`

Example view-logs.sh

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

Gestisci le applicazioni con l'API AWS Panorama

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

Visualizzazione delle applicazioni

Per ottenere un elenco di applicazioni in esecuzione su un'appliance, 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 applicativa, 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 gli stream delle videocamere

Puoi mettere in pausa e riprendere i nodi di streaming della videocamera con l'API. [SignalApplicationInstanceNodeInstances]

```
$ 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"
}
```

Mettendo in pausa e riprendendo i nodi della videocamera, è possibile scorrere un numero maggiore di stream della videocamera rispetto a quelli che possono essere elaborati contemporaneamente. A tale scopo, mappate più flussi di telecamere sullo stesso nodo di input nel manifesto di override.

Nell'esempio seguente, il manifesto di override mappa due flussi di telecamere `warehouse-floor` e lo stesso nodo `entrance-north` 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 override-multicam.json

```
    "nodeGraphOverrides": {
```

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

Per i dettagli sulla distribuzione con l'API, consulta. Automatizza 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 o AWS 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.

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

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

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

- Amazon Simple Storage Service — AWS PrivateLink per 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 in AWS 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' AWS SDK 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 VPC IDs con le tue risorse. IDs

Applicazioni, script e modelli di esempio

L' GitHub archivio di questa guida fornisce applicazioni, script e modelli di esempio per dispositivi. AWS Panorama Utilizza questi esempi per apprendere le best practice e automatizzare i flussi di lavoro di sviluppo.

Sections

Applicazioni di esempio

Le applicazioni di esempio dimostrano l'uso di AWS Panorama funzionalità 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. Utilizzatelo AWS SDK per Python (Boto) per caricare le metriche CloudWatch, strumentare i metodi di preelaborazione e inferenza e configurare la registrazione.
- debug-server: apre le porte in entrata sul 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 i modelli dal codice e compilali con SageMaker AI Neo per testare la compatibilità con l'appliance. AWS Panorama 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 una versione specifica o in TensorFlow Python.

Per altre applicazioni di esempio, visita anche il repository. aws-panorama-samples

Script di utilità

Gli script nella `util-scripts` directory gestiscono le AWS Panorama risorse o automatizzano 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 — Elimina un nodo della 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 distribuisce 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 — Aggiunge nuovamente il modello all'applicazione dopo aver aggiornato il file descrittore.
- cleanup-patches.sh: annulla la registrazione delle vecchie versioni di patch ed elimina i relativi manifest da Amazon S3.

Per i dettagli sull'utilizzo, consulta il README.

CloudFormation modelli

Utilizzate i CloudFormation modelli nella `cloudformation-templates` directory per creare risorse per AWS Panorama le applicazioni.

- 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 le metriche per 5 minuti, l'allarme invia un'email di notifica.
- application-role.yml — Crea un ruolo dell'applicazione. 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 — Crea un VPC con accesso privato al servizio di sottorete per l'appliance. AWS Panorama Per collegare l'appliance a un VPC, AWS Direct Connect utilizzare o. AWS Site-to-Site VPN
- vpc-endpoint.yml — Crea un VPC con accesso privato al servizio tramite un servizio di sottorete. AWS Panorama Le risorse all'interno del VPC possono connettersi per AWS Panorama monitorare e gestire AWS Panorama le risorse senza connettersi a Internet.

Lo `create-stack.sh` script in questa directory crea degli CloudFormation stack. 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

Il aws-panorama-samplesrepository contiene più applicazioni di esempio e strumenti utili.

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

La AWS community ha anche sviluppato strumenti e linee guida per. AWS Panorama Dai un'occhiata ai seguenti progetti open source su GitHub.

- cookiecutter-panorama — Un modello di Cookiecutter per applicazioni. AWS Panorama

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

Monitoraggio di AWS Panorama risorse e applicazioni

Puoi monitorare AWS Panorama le risorse nella AWS Panorama console e con Amazon CloudWatch. L' AWS Panorama appliance si connette al AWS cloud tramite Internet per segnalare il suo stato e lo stato delle telecamere collegate. Mentre è acceso, l'appliance invia anche i log a CloudWatch Logs in tempo reale.

L'appliance ottiene l'autorizzazione a utilizzare AWS IoT, CloudWatch Logs e altri servizi AWS da un ruolo di servizio creato la prima volta che usi la AWS Panorama console. Per ulteriori informazioni, consulta Ruoli del servizio AWS Panorama e risorse multiservizio.

Per assistenza nella risoluzione di errori specifici, consulta. Risoluzione dei problemi

Argomenti

- Monitoraggio nella console AWS Panorama
- Visualizzazione dei log di AWS Panorama
- Monitoraggio di dispositivi e applicazioni con Amazon CloudWatch

Monitoraggio nella console AWS Panorama

Puoi utilizzare la console AWS Panorama per monitorare AWS Panorama Appliance e le telecamere. La console viene utilizzata AWS IoT per monitorare lo stato dell'appliance.

Per monitorare la tua appliance nella console AWS Panorama

1. Apri la console AWS Panorama.
2. Apri la pagina Dispositivi della console AWS Panorama.
3. Scegli un'appliance.
4. Per visualizzare lo stato di un'istanza dell'applicazione, selezionala dall'elenco.
5. Per visualizzare lo stato delle interfacce di rete dell'appliance, scegliete Impostazioni.

Lo stato generale dell'appliance viene visualizzato nella parte superiore della pagina. Se lo stato è Online, l'appliance è connessa AWS e invia aggiornamenti di stato regolari.

Visualizzazione dei log di AWS Panorama

AWS Panorama riporta gli eventi delle applicazioni e dei sistemi su Amazon CloudWatch Logs. In caso di problemi, puoi utilizzare i registri degli eventi per eseguire il debug dell'applicazione AWS Panorama o risolvere i problemi di configurazione dell'applicazione.

Per visualizzare i log nei log CloudWatch

1. Apri la pagina dei gruppi di log della console CloudWatch 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`

Quando si esegue nuovamente il provisioning di un dispositivo dopo l'aggiornamento del software di sistema, è inoltre possibile visualizzare i registri sull'unità USB di provisioning.

Sections

Visualizzazione dei registri dei dispositivi

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

Registri dei dispositivi: **`/aws/panorama/devices/device-id`**

- `occ_log`— Uscita dal processo del controller. Questo processo coordina le distribuzioni delle applicazioni e genera report sullo stato dei nodi di ciascuna istanza dell'applicazione.

- `ota_log`— Output del processo che coordina gli aggiornamenti del software over-the-air (OTA).
- `syslog`— Output dal processo syslog del dispositivo, che acquisisce i messaggi inviati tra i processi.
- `kern_log`— Eventi dal kernel Linux del dispositivo.
- `logging_setup_logs`— Output del processo che configura l'agente CloudWatch Logs.
- `cloudwatch_agent_logs`— Output dall'agente CloudWatch Logs.
- `shadow_log`— Uscita dall'ombra del AWS IoT dispositivo.

Visualizzazione dei log delle applicazioni

Il gruppo di log di un'istanza dell'applicazione contiene un flusso di log per ogni nodo, che prende il nome dal nodo.

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

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

Se non vedi i log in CloudWatch Logs, conferma di trovarti nella regione AWS corretta. In tal caso, potrebbe esserci un problema con la connessione dell'appliance ad AWS o con le autorizzazioni sul ruolo dell'appliance AWS Identity and Access Management (IAM).

Configurazione dei log delle applicazioni

Configura un logger Python su cui scrivere i file di registro. `/opt/aws/panorama/logs` L'appliance trasmette i log da questa posizione a Logs. CloudWatch Per evitare di utilizzare troppo spazio su disco, utilizzate una dimensione massima del file di 10 MiB e un numero di backup pari a 1. L'esempio seguente 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 a livello globale e usalo in tutto il codice dell'applicazione.

Example application.py — Inizializza il 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 dei registri di approvvigionamento

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

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

Log di applicazioni

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

Registrazione dei log in uscita da un dispositivo

Se i log del dispositivo e dell'applicazione non vengono visualizzati in CloudWatch Registri, puoi utilizzare un'unità USB per estrarre 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 FAT32 USB formattata con almeno 1 GB di spazio di archiviazione, per il trasferimento dei file di registro da AWS Panorama Appliance.

Per registrare i log in uscita dal dispositivo

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

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

2. Connect l'unità USB al dispositivo.
3. Spegni l'AWS Panorama Appliance.

4. Accendi AWS Panorama Appliance.
5. Il dispositivo copia i log sul dispositivo. Il LED di stato lampeggia in blu durante l'operazione.
6. I file di registro possono quindi essere trovati all'interno della `managed_logs` directory con il formato `panorama_device_log_v1_dd_hh_mm.img`

Non puoi decifrare l'immagine di registro da solo. 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 Amazon CloudWatch

Quando un'appliance è online, AWS Panorama invia i parametri ad Amazon. CloudWatch Puoi creare grafici e dashboard con queste metriche nella CloudWatch console per monitorare l'attività delle appliance e impostare allarmi che ti avvisano quando i dispositivi vanno offline o le applicazioni riscontrano errori.

Per visualizzare le metriche nella console CloudWatch

1. Apri la pagina Metrics della console AWS Panorama (`PanoramaDeviceMetrics`namespace).
2. Scegli uno schema di dimensione.
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**
>
> CloudWatch ha un livello Always Free. Oltre la soglia del livello gratuito, CloudWatch addebiti per metriche, dashboard, allarmi, registri e approfondimenti. Per informazioni dettagliate, consulta Prezzi di CloudWatch .

Per ulteriori informazioni CloudWatch, consulta la Amazon CloudWatch User Guide.

Sections

- Utilizzo delle metriche dei dispositivi
- Utilizzo delle metriche delle applicazioni
- Configurazione degli allarmi

Utilizzo delle metriche dei dispositivi

Quando un'appliance è online, invia i parametri ad Amazon. CloudWatch Puoi utilizzare queste metriche per monitorare l'attività del dispositivo e attivare un allarme se i dispositivi vanno offline.

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

Dimensioni — `DeviceId` e`DeviceName`.

Visualizza la `DeviceActive` metrica con la `Average` statistica.

Utilizzo delle metriche delle applicazioni

Quando un'applicazione rileva un errore, invia i parametri ad Amazon. CloudWatch Puoi utilizzare queste metriche per attivare un allarme se un'applicazione smette di funzionare.

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

  Dimensioni — `ApplicationInstanceName` e`ApplicationInstanceId`.

Visualizza le metriche delle applicazioni con la `Sum` statistica.

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 della `ApplicationErrors` metrica rimane uguale a 1 per 20 minuti.

Per creare un allarme

1. Apri la pagina Allarmi CloudWatch della console Amazon.
2. Scegli Crea allarme.
3. Scegli Seleziona metrica e individua una metrica per il tuo dispositivo, ad esempio per,. `ApplicationErrors` `applicationInstance-gk75xmplqbqtenlnmz4ehiu7xa` `my-application`
4. Segui le istruzioni per configurare una condizione, un'azione e un nome per l'allarme.

Per istruzioni dettagliate, consulta Creare un CloudWatch allarme nella Amazon CloudWatch User Guide.

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 dei log di AWS Panorama.

Provisioning

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 Panorama appliance 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® SE7 0, l'unità USB non è inclusa nell'accessorio. Utilizza 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 Appliance

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

AWS Panorama non 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 i 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: servizio AWS Panorama

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 Camera

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 Panorama Application 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 Gestione dei segreti AWS 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 telecamera

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. Telecamere supportate

Sicurezza in AWS Panorama

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

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

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei programmi di conformitàAWS. Per ulteriori informazioni sui programmi di conformità che si applicano ad AWS Panorama, consulta AWS Services in Scope by Compliance Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal servizio AWS che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi AWS Panorama. I seguenti argomenti mostrano come configurare AWS Panorama per soddisfare i tuoi obiettivi di sicurezza e conformità. Imparerai anche a usare altri servizi AWS che ti aiutano a monitorare e proteggere le tue risorse AWS Panorama.

Argomenti

Caratteristiche di sicurezza di AWS Panorama Appliance

Per proteggere applicazioni, modelli e hardware da codice dannoso e altri exploit, AWS Panorama Appliance implementa un set completo di funzionalità di sicurezza. Queste includono, a titolo esemplificativo ma non esaustivo, quanto segue.

- Crittografia completa del disco: l'appliance implementa la crittografia completa del disco con configurazione a chiave unificata Linux ()LUKS2. Tutti i dati del software e delle applicazioni di sistema sono crittografati con una chiave specifica per il dispositivo. Anche con l'accesso fisico al dispositivo, un utente malintenzionato non può ispezionare il contenuto del suo spazio di archiviazione.
- Randomizzazione del layout della memoria: per proteggersi dagli attacchi che prendono di mira il codice eseguibile caricato in memoria, AWS Panorama Appliance utilizza la randomizzazione del layout dello spazio degli indirizzi (ASLR). ASLR rende casuale la posizione del codice del sistema operativo quando viene caricato in memoria. Ciò impedisce l'uso di exploit che tentano di sovrascrivere o eseguire sezioni specifiche di codice prevedendo dove vengono archiviate in fase di esecuzione.
- Ambiente di esecuzione affidabile: l'appliance utilizza un ambiente di esecuzione affidabile (TEE) basato su ARM TrustZone, con risorse di archiviazione, memoria ed elaborazione isolate. Le chiavi e gli altri dati sensibili archiviati nella zona di fiducia sono accessibili solo da un'applicazione affidabile, 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 effettuando 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 credenziali di breve durata per connettersi AWS IoT e richiede automaticamente un certificato valido per un periodo di tempo più lungo. Il servizio AWS Panorama genera credenziali e le crittografa con una chiave codificata sul dispositivo. Solo il dispositivo che ha richiesto il certificato può decrittografarlo e comunicare con AWS Panorama.
- Avvio sicuro: all'avvio del dispositivo, ogni componente software viene autenticato prima dell'esecuzione. La ROM di avvio, software codificato nel processore che non può essere modificato, utilizza una chiave di crittografia codificata per decrittografare il bootloader, che convalida il kernel dell'ambiente di esecuzione affidabile 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 alla convalida dei moduli del kernel, l'appliance utilizza la `dm-verity` funzionalità di Linux Device Mapper per verificare l'integrità dell'immagine del software dell'appliance prima del montaggio. Se il software dell'appliance viene modificato, non verrà eseguito.
- Prevenzione del rollback: quando si aggiorna il software dell'appliance, quest'ultima attiva un fusibile elettronico sul SoC (system on a chip). Ogni versione del software prevede che si bruci un numero crescente di fusibili e non può funzionare se ne vengono bruciati altri.

Best practice di sicurezza per AWS Panorama Appliance

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

- Proteggi fisicamente l'appliance: installa l'appliance in un server rack chiuso o in una stanza sicura. Limita l'accesso fisico al dispositivo al personale autorizzato.
- Proteggi la connessione di rete dell'appliance: collega l'appliance a un router che limita l'accesso alle risorse interne ed esterne. L'apparecchiatura deve connettersi alle telecamere, che possono trovarsi su una rete interna sicura. È inoltre necessario connettersi a. AWS Utilizza la seconda porta Ethernet solo per la ridondanza fisica e configura il router per consentire solo il traffico richiesto.

  Utilizza una delle configurazioni di rete consigliate per pianificare il layout della rete. Per ulteriori informazioni, consulta Connessione di AWS Panorama Appliance alla rete.
- Formattazione dell'unità USB: dopo aver fornito un dispositivo, rimuovi l'unità USB e formattala. L'appliance non utilizza l'unità USB dopo la registrazione con il servizio AWS Panorama. Formatta l'unità per rimuovere credenziali temporanee, file di configurazione e registri di provisioning.
- Mantieni l'appliance aggiornata: applica gli aggiornamenti software dell'appliance in modo tempestivo. Quando visualizzi un'appliance nella console AWS Panorama, la console ti avvisa se è disponibile un aggiornamento software. Per ulteriori informazioni, consulta Gestione di un'appliance AWS Panorama.

  Con il funzionamento dell'DescribeDeviceAPI, puoi automatizzare il controllo degli aggiornamenti confrontando i campi e. `LatestSoftware` `CurrentSoftware` Quando la versione più recente del software è diversa dalla versione corrente, applica l'aggiornamento con la console o utilizzando l'CreateJobForDevicesoperazione.
- Se smetti di usare un dispositivo, ripristinalo: prima di spostare l'appliance dal tuo data center sicuro, ripristinala completamente. Con l'apparecchio spento e collegato, premi contemporaneamente il pulsante di accensione e il pulsante di ripristino per 5 secondi. Ciò elimina le credenziali dell'account, le applicazioni e i registri dall'appliance.

  Per ulteriori informazioni, consulta Pulsanti e luci di AWS Panorama Appliance.
- Limita l'accesso ad AWS Panorama e ad altri servizi AWS: AWSPanoramaFullAccessfornisce l'accesso a tutte le operazioni dell'API AWS Panorama e, se necessario, l'accesso ad altri servizi. Ove possibile, la policy limita l'accesso alle risorse in base a convenzioni di denominazione. Ad esempio, fornisce l'accesso ai Gestione dei segreti AWS segreti i cui nomi iniziano con. `panorama`

Per gli utenti che necessitano dell'accesso in sola lettura o dell'accesso a un set di risorse più specifico, utilizza la politica gestita come punto di partenza per le politiche con privilegi minimi.

Per ulteriori informazioni, consulta Policy IAM basate sull'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 gestisce tutto il Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. 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 .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or 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.
- Usa 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 Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno 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 hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo 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 AWS CLI, l'API o AWS SDKs. 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' Console di gestione AWS SDK 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'appliance AWS Panorama 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 archiviate Gestione dei segreti AWS. 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 software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse AWS Panorama. IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

Argomenti

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all’amministratore se non riesci ad accedere alle funzionalità (consulta Risoluzione dei problemi di identità e accesso ad AWS Panorama)
- Amministratore del servizio: determina l’accesso degli utenti e invia le richieste di autorizzazione (consulta Come funziona AWS Panorama con IAM)
- Amministratore IAM: scrivi policy per gestire l’accesso (consulta Esempi di policy basate sull'identità di AWS Panorama)

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/ Facebook Per ulteriori informazioni sull'accesso, consulta Come accedere all' Account AWS nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta AWS Signature Version 4 per le richieste API nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali come utente root, consulta 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 che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella Guida per l'utente IAM.

Un gruppo IAM specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente di IAM.

Ruoli IAM

Un ruolo IAM è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo passando da un ruolo utente a un ruolo IAM (console) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta Metodi per assumere un ruolo nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Gestione dell’accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l’utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull’identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un’identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l’identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l’utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scegliere tra policy gestite e policy in linea nella Guida per l’utente di IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l’accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio specificare un’entità principale.

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

Liste di controllo degli accessi () ACLs

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

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta Politiche di controllo delle risorse (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

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 AWS i servizi che funzionano con IAM nella IAM User Guide.

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

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 possono eseguire attività utilizzando l' AWS API Console di gestione AWS AWS CLI, o. 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 azioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo

definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori 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 ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare Protezione dell'accesso API 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, consulta Policy IAM basate sull'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 collegate alla relativa identità utente. Questa policy include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o in modo programmatico. AWS CLI 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": "*"
        }
    ]
}
```

AWS politiche gestite per AWS Panorama

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

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti 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 politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consulta Policy gestite da AWSnella 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 delle appliance e ai log delle appliance in Gestione dei segreti AWS Amazon. CloudWatch Include l'autorizzazione a creare un ruolo collegato ai servizi per 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 log e ottenere oggetti dai punti di accesso Amazon S3 creati da AWS Panorama. CloudWatch

Aggiornamenti di AWS Panorama alle policy AWS gestite

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

Modifica	Descrizione	Data
AWSPanoramaApplianceServiceRolePolicy — Aggiornamento a una policy esistente	Sostituisci StringLike la condizione con ArnLike per la scrittura ARNs.	2024-12-10
AWSPanoramaFullAccess — Aggiornamento a una politica esistente	Sostituisci StringLike la condizione con ArnLike per la scrittura ARNs.	2024-12-10
AWSPanoramaFullAccess — Aggiornamento a una politica esistente	Sono state aggiunte autorizzazioni alla politica utente per consentire agli utenti di visualizzare i gruppi di log nella console CloudWatch Logs.	13/01/2022
AWSPanoramaFullAccess — Aggiornamento a una politica esistente	Sono state aggiunte autorizzazioni alla policy utente per consentire agli utenti di gestire il ruolo collegato al servizio AWS Panorama e di accedere alle risorse AWS Panorama in altri servizi tra cui IAM, Amazon S3 e CloudWatch Secrets Manager.	2021-10-20
AWSPanoramaApplianceServiceRolePolicy — Nuova politica	Nuova policy per il ruolo del servizio AWS Panorama Appliance	2021-10-20
AWSPanoramaServiceLinkedRolePolicy — Nuova politica	Nuova policy per il ruolo collegato al servizio AWS Panorama.	2021-10-20

Modifica	Descrizione	Data
AWS Panorama ha iniziato a tracciare le modifiche	AWS Panorama ha iniziato a tracciare le modifiche per le sue policy AWS gestite.	2021-10-20

Utilizzo di ruoli collegati ai servizi per AWS Panorama

AWS Panorama utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente. AWS Panorama I ruoli collegati ai servizi sono predefiniti AWS Panorama e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione AWS Panorama perché non è necessario aggiungere manualmente le autorizzazioni necessarie. AWS Panorama definisce le autorizzazioni dei ruoli collegati ai servizi e, se non diversamente definito, solo può assumerne i ruoli. AWS Panorama 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, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Yes (Sì) nella colonna Service-linked roles (Ruoli collegati ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Sections

- Autorizzazioni di ruolo collegate al servizio per AWS Panorama
- Creazione di un ruolo collegato al servizio per AWS Panorama
- Modifica di un ruolo collegato al servizio per AWS Panorama
- Eliminazione di un ruolo collegato al servizio per AWS Panorama
- Regioni supportate per i ruoli collegati ai servizi AWS Panorama

Autorizzazioni di ruolo collegate al servizio per AWS Panorama

AWS Panorama utilizza il ruolo collegato al servizio denominato AWSServiceRoleForAWSPanorama: consente ad AWS Panorama di gestire le risorse in AWS IoT, AWS Secrets Manager e AWS Panorama.

Il ruolo AWSService RoleFor AWSPanorama collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `panorama.amazonaws.com`

La politica di autorizzazione dei ruoli consente di AWS Panorama completare le seguenti azioni:

- Monitora le risorse AWS Panorama
- Gestisci AWS IoT le risorse per l' AWS Panorama appliance
- Accedi ai Gestione dei segreti AWS segreti per ottenere le credenziali della fotocamera

Per un elenco completo delle autorizzazioni, visualizza la AWSPanorama ServiceLinkedRolePolicy policy nella 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 al servizio per AWS Panorama

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando registri un'appliance nella Console di gestione AWS, la o l' AWS API AWS CLI, AWS Panorama crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato ai servizi, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando registrate un'appliance, AWS Panorama crea nuovamente il ruolo collegato al servizio.

Modifica di un ruolo collegato al servizio per AWS Panorama

AWS Panorama non consente di modificare il ruolo collegato al AWSService RoleFor AWSPanorama servizio. Dopo aver creato un ruolo collegato al servizio, 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 al servizio per AWS Panorama

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, 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 al servizio prima di poterlo eliminare manualmente.

Per eliminare le AWS Panorama risorse utilizzate da AWSService RoleForAWSPanorama, utilizzare le procedure descritte nelle sezioni seguenti di questa guida.

- Eliminare versioni e applicazioni
- Annulla la registrazione di un dispositivo

> **Note**
>
> Se il AWS Panorama servizio utilizza il ruolo quando si tenta di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare il ruolo AWSService RoleFor AWSPanorama collegato al servizio, utilizza la console IAM AWS CLI, o l' AWS API. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio 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 al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta AWS Regioni ed endpoint.

Prevenzione del confused deputy tra servizi

Il problema confused deputy è un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire un'azione può costringere un'entità maggiormente privilegiata a eseguire l'azione. Nel AWS, l'impersonificazione tra servizi può portare al confuso problema del vice. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua

una chiamata a 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 che ciò accada, AWS mette a disposizione strumenti che consentono di proteggere i dati relativi a tutti i servizi con responsabili del servizio a cui è stato concesso l'accesso alle risorse del vostro account.

Ti consigliamo di utilizzare `aws:SourceArn`le chiavi di contesto della condizione `aws:SourceAccount`globale nelle politiche delle risorse per limitare le autorizzazioni che AWS Panorama forniscono 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 un AWS Panorama dispositivo.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non si conosce l'ARN completo della risorsa o si scelgono più risorse, è necessario utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (`*`) per le parti sconosciute dell'ARN. Ad esempio `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

Per istruzioni su come proteggere il ruolo di servizio AWS Panorama utilizzato per concedere l'autorizzazione all' AWS Panorama Appliance, vedere. Garantire 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 ti Console di gestione AWS dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

L'errore di esempio seguente 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 chiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa `my-appliance` mediante 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 AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. 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 trasmettere 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`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. 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 i servizi che supportano policy basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali policy per concedere alle persone l'accesso alle tue risorse.

Per maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS Panorama supporta queste funzionalità, consultaCome funziona AWS Panorama con IAM.
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti 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 IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Convalida della conformità per AWS Panorama

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program Servizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

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

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta AWS la documentazione sulla sicurezza.

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 un avviso 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 dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizza chiamate API AWS pubblicate per accedere ad AWS Panorama attraverso la rete. I client 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.

Implementazione di AWS Panorama Appliance nel tuo datacenter

L'AWS Panorama Appliance necessita dell'accesso a Internet per comunicare con AWS i servizi. Inoltre, necessita dell'accesso 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 l'appliance AWS Panorama 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.

L'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 dettagli sulla crittografia, consultaProtezione dei dati in AWS Panorama.

Software di ambiente di 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 ha la responsabilità di mantenere aggiornato il software nell'immagine dell'appliance. AWS Panorama rilascia regolarmente aggiornamenti software, che puoi applicare utilizzando la console AWS Panorama.

Puoi utilizzare le librerie nel codice dell'applicazione installandole nell'applicazione`Dockerfile`. Per garantire la stabilità dell'applicazione tra le build, scegliete una versione specifica di ogni 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 Panorama appliance alla versione più recente del software. Per ulteriori informazioni su una versione, consultate l'argomento collegato.

Modifica	Descrizione	Data
Avviso di fine del supporto	Avviso di fine del supporto: il 31 maggio 2026, AWS terminerà il supporto per AWS Panorama. Dopo il 31 maggio 2026, non potrai più accedere alla AWS Panorama console o AWS Panorama alle risorse. Per ulteriori informazioni, consulta AWS Panorama Fine del supporto.	20 maggio 2025
Politiche gestite aggiornate	AWS Identity and Access Management le politiche gestite per sono AWS Panorama state aggiornate. Per i dettagli, consulta le policy gestite da AWS.	10 dicembre 2024
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	28 dicembre 2023

	aggiuntivi prima di applicare l'aggiornamento. Per ulteriori informazioni, consulta il registro delle modifiche.	
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 enti nella gestione degli errori. Per ulteriori informazi oni, consulta il registro delle modifiche.	31 marzo 2023
Aggiornamento della console	È ora possibile acquistare l' AWS Panorama appliance dalla console di gestione. Per concedere a un utente l'autoriz zazione all'acquisto di dispositi vi, 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 miglioram enti nella gestione degli errori. Per ulteriori informazi oni, 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, consulta 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
AWS PrivateLink supporto	AWS Panorama supporta 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® SE7 0	Un nuovo dispositivo per AWS Panorama è disponibile presso Lenovo. Il Lenovo ThinkEdge ® SE7 0, basato su Nvidia Jetson Xavier NX, supporta le stesse funzionalità dell'appl iance. 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'esecuzi one 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 Management le 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 Panorama appliance 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 Panorama è ora disponibile 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 Management le politiche gestite per sono AWS Panorama state aggiornate. Per i dettagli, consulta le policy gestite da AWS.	20 ottobre 2021
Disponibilità generale	AWS Panorama è ora disponibile per tutti i clienti nelle regioni Stati Uniti orientali (Virginia settentri onale), Stati Uniti occidenta li (Oregon), Europa (Irlanda) e Canada (Centrale). Per acquistare un AWS Panorama elettrodomestico, visita. AWS Panorama	20 ottobre 2021
Anteprima	AWS Panorama è disponibi le su invito nelle regioni Stati Uniti orientali (Virginia settentri onale) e Stati Uniti occidentali (Oregon).	1 dicembre 2020