aws

Guía para desarrolladores

AWS Panorama

AWS Panorama: Guía para desarrolladores

Table of Contents

Aviso de fin de soporte: el 31 de mayo de 2026, AWS finalizará el soporte para AWS Panorama. Después del 31 de mayo de 2026, ya no podrás acceder a la AWS Panorama consola ni a AWS Panorama los recursos. Para obtener más información, consulta AWS Panorama el fin del soporte.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.

¿Qué es AWS Panorama?

AWS Panorama es un servicio que lleva la visión artificial a su red de cámaras local. Instale el AWS Panorama dispositivo u otro dispositivo compatible en su centro de datos, lo registre e implemente aplicaciones de visión artificial desde la nube. AWS Panorama AWS Panorama funciona con sus cámaras de red con protocolo de transmisión en tiempo real (RTSP) existentes. El dispositivo ejecuta aplicaciones de visión artificial seguras de AWS nuestros socios o aplicaciones que usted mismo ha creado con el SDK de AWS Panorama aplicaciones.

El AWS Panorama dispositivo es un dispositivo periférico compacto que utiliza un potente system-on-module (SOM) optimizado para las cargas de trabajo de aprendizaje automático. El dispositivo puede ejecutar varios modelos de visión artificial en múltiples flujos de vídeo en paralelo y emitir los resultados en tiempo real. Está diseñado para su uso en entornos comerciales e industriales y tiene una clasificación de protección contra polvo y líquidos (IP-62).

El AWS Panorama dispositivo le permite ejecutar aplicaciones de visión artificial independientes en la periferia, sin enviar imágenes a la nube de AWS. Con el SDK de AWS, puede integrarse con otros servicios de AWS y utilizarlos para realizar un seguimiento de los datos de la aplicación a lo largo del tiempo. Al integrarlo con otros servicios de AWS, puede AWS Panorama hacer lo siguiente:

- Analizar los patrones de tráfico: utilice el SDK de AWS para registrar datos para el análisis de ventas minoristas en Amazon DynamoDB. Utilice una aplicación sin servidor para analizar los datos recopilados a lo largo del tiempo, detectar anomalías en los datos y predecir el comportamiento futuro.
- Recibir alertas de seguridad en las instalaciones: supervise las áreas prohibidas de una planta industrial. Cuando su aplicación detecte una situación potencialmente no segura, suba una imagen a Amazon Simple Storage Service (Amazon S3) y envíe una notificación a un tema de Amazon Simple Notification Service (Amazon SNS) para que los destinatarios puedan realizar cambios en el tema.
- Mejorar el control de calidad: supervise la producción de una línea de montaje para identificar las piezas que no cumplen con los requisitos. Resalte las imágenes de las piezas no conformes con texto y un recuadro delimitador y muéstrelas en un monitor para que su equipo de control de calidad las revise.
- Recopilar datos de entrenamiento y pruebas: cargue imágenes de objetos que su modelo de visión artificial no haya podido identificar o en los que el modelo no pueda confiar en su suposición. Use

una aplicación sin servidor para crear una cola de imágenes que deban etiquetarse. Etiquete las imágenes y utilícelas para volver a entrenar el modelo en Amazon SageMaker AI.

AWS Panorama utiliza otros servicios de AWS para administrar el AWS Panorama dispositivo, acceder a modelos y códigos e implementar aplicaciones. AWS Panorama hace todo lo posible sin necesidad de que interactúe con otros servicios, pero conocer los siguientes servicios puede ayudarle a entender cómo AWS Panorama funcionan.

- SageMaker IA: puedes usar la SageMaker IA para recopilar datos de entrenamiento de cámaras o sensores, crear un modelo de aprendizaje automático y entrenarlo para la visión artificial. AWS Panorama utiliza SageMaker AI Neo para optimizar los modelos para que se ejecuten en el AWS Panorama dispositivo.
- Amazon S3: utiliza los puntos de acceso de Amazon S3 para organizar el código de la aplicación, los modelos y los archivos de configuración para su implementación en un AWS Panorama dispositivo.
- AWS IoT— AWS Panorama utiliza AWS IoT servicios para supervisar el estado del AWS Panorama dispositivo, gestionar las actualizaciones de software e implementar aplicaciones. No necesita usarlo AWS IoT directamente.

Para empezar a utilizar el AWS Panorama dispositivo y obtener más información sobre el servicio, continúe conEmpezar con AWS Panorama.

Fin del soporte de AWS Panorama

Tras considerarlo detenidamente, decidimos poner fin al soporte para el Panorama de AWS a partir del 31 de mayo de 2026. AWS Panorama ya no aceptará nuevos clientes a partir del 20 de mayo de 2025. Como cliente actual con una cuenta registrada en el servicio antes del 20 de mayo de 2025, puede seguir utilizando las funciones de AWS Panorama. Después del 31 de mayo de 2026, ya no podrá utilizar AWS Panorama.

Alternativas a AWS Panorama

Si está interesado en una alternativa a AWS Panorama, AWS tiene opciones tanto para compradores como para desarrolladores.

Para out-of-the-box encontrar una solución, la red de socios de AWS ofrece soluciones de varios socios. Puede buscar soluciones de muchos de nuestros socios en la biblioteca de soluciones de AWS. Estas soluciones de socios incluyen opciones de hardware, software, aplicaciones de software como servicio (SaaS), soluciones administradas o implementaciones personalizadas en función de sus necesidades. Este enfoque proporciona una solución que se adapta a su caso de uso sin necesidad de que tenga experiencia en visión artificial, inteligencia artificial o desarrollo de aplicaciones. Por lo general, esto permite rentabilizar más rápidamente al aprovechar la experiencia especializada de los socios de AWS.

Si prefiere crear su propia solución, AWS ofrece herramientas y servicios de IA que le ayudarán a desarrollar una aplicación de visión artificial basada en la IA y a gestionar las aplicaciones y los dispositivos periféricos. Amazon SageMaker proporciona un conjunto de herramientas para crear, entrenar e implementar modelos de aprendizaje automático para su caso de uso con una infraestructura, herramientas y flujos de trabajo totalmente administrados. Además de permitirte crear tus propios modelos, Amazon SageMaker JumpStart ofrece algoritmos de visión artificial integrados que se pueden ajustar a tu caso de uso específico.

Para administrar dispositivos y aplicaciones en la periferia, AWS IoT Greengrass es una solución comprobada y segura para implementar y actualizar aplicaciones para dispositivos de IoT. Para una implementación basada en servidores, AWS Systems Manager proporciona un conjunto de herramientas para administrar servidores y Amazon EKS Anywhere o ECS Anywhere pueden administrar contenedores de aplicaciones en servidores perimetrales. Amazon proporciona algunas pautas para la administración de dispositivos periféricos, junto con recursos adicionales, en la sección 4 del documento técnico Cómo proteger el Internet de las cosas (IoT) con AWS. Este enfoque de creación le proporciona las herramientas necesarias para acelerar el desarrollo de la IA y la administración de dispositivos y, al mismo tiempo, le proporciona una flexibilidad total para crear una solución que satisfaga sus requisitos exactos y se integre con su infraestructura de hardware y software existente. Por lo general, esto reduce los costos operativos de una solución.

Migración desde AWS Panorama

Para migrar una aplicación existente de AWS Panorama a una implementación alternativa, tendrá que reemplazar el dispositivo de hardware existente, migrar la aplicación desde el servicio AWS Panorama e implementar la administración perimetral y la seguridad para la nueva solución. Cada una de estas áreas se analizará en detalle a continuación:

Reemplazo de hardware

El dispositivo AWS Panorama existente se basa en la plataforma Nvidia Jetson Xavier. El hardware se puede reemplazar por un off-the-shelf dispositivo similar basado en la plataforma Nvidia Jetson de la generación actual que cumpla con sus requisitos, o por un servidor perimetral. Si bien la mayoría de las implementaciones de AWS Panorama se pueden reemplazar por un dispositivo similar, hemos visto que algunos clientes que utilizan una gran cantidad de cámaras en una sola ubicación consideran que un servidor es una mejor alternativa.

Migración de aplicaciones

Las aplicaciones de AWS Panorama deben reescribirse para eliminar el uso de llamadas a la API específicas de AWS Panorama. Las aplicaciones de AWS Panorama solo admiten la entrada de vídeo a través del Protocolo de transmisión en tiempo real (RTSP) mediante H.264 y esas entradas de vídeo se proporcionan mediante los nodos de cámara del SDK del dispositivo AWS Panorama.

Para migrar una aplicación existente, necesitará implementar una clase de aplicación similar a AWS Panorama para que el código existente pueda reutilizarse en su mayor parte. El código de muestra está disponible en el archivo banner-code.zip, que muestra un ejemplo de esta implementación con PyAV y OpenCV.

Se trata de un enfoque sencillo con una cantidad mínima de cambios en el código, pero tiene muchas de las mismas limitaciones que la implementación actual basada en el Panorama de AWS en cuanto a los tipos de transmisiones de vídeo compatibles.

Otra opción sería rediseñar la aplicación para aprovechar mejor los recursos del sistema y admitir las nuevas capacidades de la aplicación. Para esta opción, puede utilizar GStreamero DeepStreamimplementar el proceso multimedia, desde la fuente multimedia hasta los resultados de inferencia y la lógica empresarial, o bien utilizar una implementación de tiempo de ejecución de aprendizaje automático (ML) con más funciones y mejor rendimiento, como el servidor de inferencia Nvidia Triton. Este enfoque requiere cambios en una mayor parte del proceso de procesamiento de vídeo, pero es más eficiente y ofrece una mayor flexibilidad para admitir una gama más amplia de códecs, tipos de cámaras y otros sensores.

Gestión perimetral y seguridad

Independientemente del canal multimedia, también tendrá que implementar un almacén seguro de credenciales, por ejemplo, el nombre de usuario y la contraseña de la transmisión RTSP. AWS proporciona diferentes formas de almacenar de forma segura los parámetros de las aplicaciones:

- El servicio AWS IoT Device Shadow se utiliza para almacenar los parámetros que se transfieren a las aplicaciones, así como para realizar un seguimiento del estado de las aplicaciones en el dispositivo perimetral.
- AWS Secrets Manager se utiliza para almacenar dichas credenciales a fin de proteger mejor las credenciales de acceso a las transmisiones multimedia.
- Si utiliza Amazon EKS o Amazon ECS, también puede utilizar el almacén seguro de parámetros de AWS System Manager para las credenciales y otros parámetros de la aplicación.

La elección depende de los requisitos de seguridad de la aplicación, así como de AWS los otros productos que vaya a utilizar para implementarla.

Al reemplazar el dispositivo AWS Panorama por un dispositivo perimetral genérico, también debe implementar las funciones de seguridad necesarias para sus aplicaciones y configurar los dispositivos para que cumplan con sus requisitos de seguridad. AWS proporciona orientación al respecto en el pilar de seguridad del AWS Well-Architected Framework. Si bien el marco se centra principalmente en las aplicaciones en la nube, la mayoría de los principios también se aplican a los dispositivos periféricos. Además, debe utilizar las funciones de seguridad de hardware de la solución elegida, como la integración de seguridad de hardware de AWS IoT Greengrass V2, y utilizar las funciones de seguridad proporcionadas por el sistema operativo o dispositivo elegido, como el cifrado de disco completo.

Resumen

Aunque AWS Panorama planea cerrar el 31 de mayo de 2026, AWS ofrece un potente conjunto de servicios y soluciones de IA/ML en forma de SageMaker herramientas de Amazon para crear modelos de visión artificial y servicios de administración de dispositivos, como AWS IoT Greengrass, Amazon EKS y Amazon ECS Anywhere y AWS System Manager para respaldar el desarrollo de soluciones similares. AWS también cuenta con una gama de ofertas de socios de la red de socios de AWS si prefiere comprar una solución en lugar de crearla. Se proporcionan ejemplos de código e implementación para ayudarlo a migrar a una solución alternativa si así lo desea. Debe explorar estas opciones para determinar qué es lo que mejor se adapta a sus necesidades específicas.

Para obtener más información, consulte los siguientes recursos:

- Guía para SageMaker desarrolladores de Amazon: documentación detallada sobre cómo crear un modelo o trabajar con algoritmos de visión artificial integrados disponible en SageMaker JumpStart.

- Guía para desarrolladores de AWS IoT Core: documentación detallada sobre cómo conectar y administrar dispositivos de IoT.
- Guía para desarrolladores de AWS IoT Greengrass V2: documentación detallada sobre cómo crear, implementar y administrar aplicaciones de IoT en sus dispositivos.
- Guía para desarrolladores de ECS Anywhere: documentación detallada sobre la ejecución de ECS en entornos periféricos.
- Guía de mejores prácticas de EKS Anywhere: documentación detallada sobre cómo ejecutar EKS en la periferia.
- Biblioteca de soluciones de AWS: ofertas de socios de una variedad de proveedores que ofrecen soluciones de visión artificial prediseñadas o personalizadas.
- Panorama FAQs: información panorámica adicional.

Preguntas frecuentes

¿En qué momento se suspende el Panorama?

El anuncio se hizo el 20 de mayo de 2025. Después de esta fecha, los clientes que no estén activos en el servicio dejarán de tener acceso a Panorama. Los clientes activos podrán seguir utilizando el servicio con normalidad hasta el 31 de mayo de 2026. Los clientes tienen hasta ese momento para trasladar su aplicación a una solución alternativa y migrar las aplicaciones de Panorama. Después del 31 de mayo de 2026, cualquier aplicación que intente acceder al servicio Panorama dejará de funcionar y los dispositivos Panorama dejarán de funcionar.

¿Cómo se verán afectados los clientes actuales?

Los clientes actuales pueden seguir utilizando el servicio con normalidad hasta el 31 de mayo de 2026. Después de eso, las aplicaciones que intenten acceder a Panorama dejarán de funcionar. Los dispositivos Panorama tampoco funcionarán después de esa fecha.

¿Se aceptan nuevos clientes?

¿No?. A partir del 20 de mayo de 2025, solo los clientes que sean usuarios activos de Panorama tendrán acceso al servicio. Si un cliente tiene aplicaciones del servicio relacionadas con un uso anterior a las que necesita acceder, puede crear un caso con el servicio de atención al cliente para solicitar el acceso a su cuenta. Si un cliente no ha utilizado el servicio anteriormente, no se le concederá el acceso.

¿Cuáles son las alternativas que pueden explorar los clientes?

AWS ofrece una gama de servicios que pueden reemplazar las capacidades de Panorama. Recomendamos a los clientes que utilicen el off-the-shelf hardware y administren el dispositivo y la aplicación mediante la combinación de AWS IoT Core, AWS IoT Greengrass, Amazon AKS Anywhere, Amazon ECS Anywhere y/o AWS System Manager según sus necesidades. La red de socios de AWS también tiene varias soluciones disponibles de socios con experiencia específica en Computer Visions que los clientes pueden considerar.

¿Cómo pueden los clientes migrar fuera de Panorama?

Las aplicaciones de Panorama deben modificarse para eliminar cualquier dependencia de las aplicaciones específicas de Panorama APIs, que se relacionan principalmente con la conexión de la cámara y la transmisión. AWS ha proporcionado un ejemplo de código para mostrar cómo realizar estos cambios. Una vez eliminadas esas dependencias, la aplicación se puede mover a una plataforma de hardware alternativa.

Si tengo problemas el 20 de mayo de 2025 o después de esa fecha, ¿qué asistencia estará disponible?

AWS continuará brindando soporte a Panorama hasta el final del período de notificación de descontinuación (31 de mayo de 2026). Para cualquier requisito de soporte, los clientes deben presentar un caso de soporte a través de sus canales de soporte habituales. AWS proporcionará actualizaciones de seguridad, correcciones de errores y mejoras de disponibilidad.

No puedo migrar antes del 31 de mayo de 2026. ¿Se puede extender la fecha?

Estamos seguros de que las alternativas disponibles para Panorama permitirán a los clientes migrar a una solución alternativa antes del 31 de mayo de 2026 y no tenemos previsto ampliar la disponibilidad del servicio más allá de esa fecha.

¿Seguirá funcionando mi aplicación perimetral una vez finalizado el servicio?

No. El dispositivo y las aplicaciones de Panorama dependen de la conectividad con el servicio en la nube de Panorama. Una vez que el servicio se interrumpa el 31 de mayo de 2026, ni la aplicación Panorama ni el dispositivo Panorama seguirán funcionando.

Empezar con AWS Panorama

Para empezar AWS Panorama, primero conozca los conceptos del servicio y la terminología que se utilizan en esta guía. A continuación, puede utilizar la AWS Panorama consola para registrar su AWS Panorama dispositivo y crear una aplicación. En aproximadamente una hora, podrá configurar el dispositivo, actualizar su software e implementar una aplicación de muestra. Para completar los tutoriales de esta sección, utilice el AWS Panorama dispositivo y una cámara que transmita vídeo a través de una red local.

> **Note**
>
> Para comprar un AWS Panorama dispositivo, visite la AWS Panorama consola.

La aplicación AWS Panorama de ejemplo muestra el uso de AWS Panorama las funciones. Incluye un modelo que se ha entrenado con SageMaker IA y un código de muestra que utiliza el SDK de la AWS Panorama aplicación para ejecutar inferencias y generar vídeo. La aplicación de muestra incluye una AWS CloudFormation plantilla y scripts que muestran cómo automatizar los flujos de trabajo de desarrollo e implementación desde la línea de comandos.

En los dos últimos temas de este capítulo se detallan los requisitos de los modelos y las cámaras, así como las especificaciones de hardware del dispositivo de AWS Panorama. Si aún no ha adquirido un dispositivo y cámaras, o planea desarrollar sus propios modelos de visión artificial, consulte primero estos temas para obtener más información.

Temas

Conceptos de AWS Panorama

En AWS Panorama, puede crear aplicaciones de visión artificial y desplegarlas en el dispositivo de AWS Panorama o en un dispositivo compatible para analizar las transmisiones de vídeo de las cámaras de red. Usted escribe el código de la aplicación en Python y crea contenedores de aplicaciones con Docker. Utiliza la CLI de la aplicación AWS Panorama para importar modelos de machine learning desde Amazon Simple Storage Service (Amazon S3). Las aplicaciones utilizan el SDK de la aplicación AWS Panorama para recibir entradas de vídeo de una cámara e interactuar con un modelo.

Conceptos

- El dispositivo de AWS Panorama
- Dispositivos compatibles
- Aplicaciones
- Nodos
- Modelos de

El dispositivo de AWS Panorama

El dispositivo de AWS Panorama es el hardware que ejecuta sus aplicaciones. Utilice la consola de AWS Panorama para registrar un dispositivo, actualizar su software e implementar aplicaciones en él. El software del dispositivo de AWS Panorama se conecta a las transmisiones de las cámaras, envía fotogramas de vídeo a la aplicación y muestra la salida de vídeo en una pantalla conectada.

El dispositivo de AWS Panorama es un dispositivo periférico con tecnología Nvidia Jetson AGX Xavier. En lugar de enviar imágenes a la AWS nube para su procesamiento, ejecuta las aplicaciones localmente en un hardware optimizado. Esto le permite analizar el vídeo en tiempo real y procesar los resultados de forma local. El dispositivo necesita una conexión a Internet para informar de su estado, cargar registros y realizar actualizaciones e implementaciones de software.

Para obtener más información, consulte Administración del AWS Panorama dispositivo.

Dispositivos compatibles

Además del dispositivo AWS Panorama, AWS Panorama admite dispositivos compatibles de los AWS socios. Los dispositivos compatibles admiten las mismas funciones que el dispositivo de

AWS Panorama. Registra y administra los dispositivos compatibles con la consola y la API de AWS Panorama, y crea e implementa aplicaciones de la misma manera.

- Lenovo ThinkEdge® SE7 0: con tecnología Nvidia Jetson Xavier NX

El contenido y las aplicaciones de muestra de esta guía se han desarrollado con el dispositivo de AWS Panorama. Para obtener más información sobre las características específicas de hardware y software de su dispositivo, consulte la documentación del fabricante.

Aplicaciones

Las aplicaciones se ejecutan en el dispositivo de AWS Panorama para realizar tareas de visión artificial en transmisiones de vídeo. Puede crear aplicaciones de visión artificial combinando código Python y modelos de machine learning e implementarlas en el dispositivo de AWS Panorama a través de Internet. Las aplicaciones pueden enviar vídeo a una pantalla o utilizar el SDK de AWS para enviar los resultados a los servicios de AWS.

Para crear e implementar aplicaciones, utilice la CLI de la aplicación AWS Panorama. La CLI de la aplicación AWS Panorama es una herramienta de línea de comandos que genera carpetas de aplicaciones y archivos de configuración predeterminados, crea contenedores con Docker y carga activos. Puede ejecutar varias aplicaciones en un dispositivo.

Para obtener más información, consulte Gestión de AWS Panorama aplicaciones.

Nodos

Una aplicación consta de varios componentes denominados nodos, que representan entradas, salidas, modelos y código. Un nodo puede ser solo de configuración (entradas y salidas) o incluir artefactos (modelos y código). Los nodos de código de una aplicación se agrupan en paquetes de nodos que se cargan en un punto de acceso de Amazon S3, desde donde el dispositivo de AWS Panorama puede acceder a ellos. Un manifiesto de aplicación es un archivo de configuración que define las conexiones entre los nodos.

Para obtener más información, consulte Nodos de aplicación.

Modelos de

Un modelo de visión artificial es una red de machine learning que está entrenada para procesar imágenes. Los modelos de visión artificial pueden realizar diversas tareas, como la clasificación, la

detección, la segmentación y el seguimiento. Un modelo de visión artificial toma una imagen como entrada y genera información sobre la imagen o los objetos de la imagen.

AWS Panorama admite modelos creados con PyTorch Apache MXNet y TensorFlow. Puede crear modelos con Amazon SageMaker AI o en su entorno de desarrollo. Para obtener más información, consulte ???.

Configuración del dispositivo de AWS Panorama

Para empezar a utilizar su dispositivo de AWS Panorama o dispositivo compatible, regístrelo en la consola de AWS Panorama y actualice su software. Durante el proceso de configuración, debe crear un recurso de dispositivo en AWS Panorama que represente el dispositivo físico y copiar los archivos en el dispositivo con una unidad USB. El dispositivo utiliza estos certificados y archivos de configuración para conectarse al servicio AWS Panorama. A continuación, utilice la consola de AWS Panorama para actualizar el software del dispositivo y registrar las cámaras.

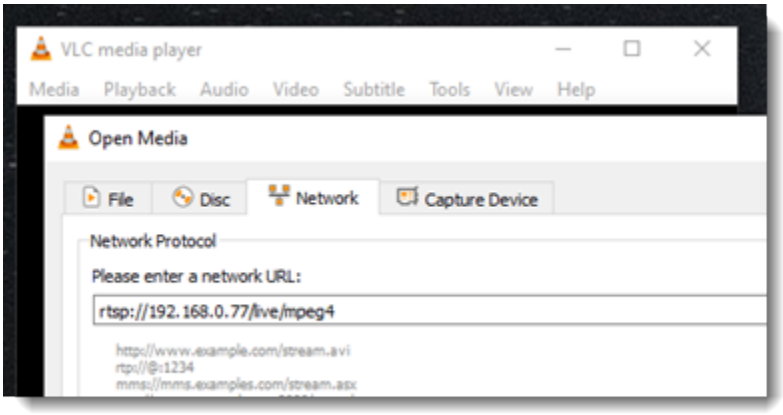
Secciones

Requisitos previos

Para seguir este tutorial, necesita un dispositivo de AWS Panorama o un dispositivo compatible y el siguiente hardware:

- Pantalla: una pantalla con entrada HDMI para ver la salida de la aplicación de muestra.
- Unidad USB (incluida con AWS Panorama Appliance): unidad de memoria flash USB 3.0 FAT32 formateada con al menos 1 GB de almacenamiento, para transferir un archivo con archivos de configuración y un certificado al dispositivo AWS Panorama.
- Cámara: una cámara IP que emite una transmisión de vídeo RTSP.

Utilice las herramientas e instrucciones proporcionadas por el fabricante de la cámara para identificar la dirección IP y la ruta de transmisión de la cámara. Puede usar un reproductor de vídeo como VLC para verificar la URL de la transmisión abriéndola como una fuente multimedia de red:



La consola de AWS Panorama usa otros servicios de AWS para ensamblar los componentes de la aplicación, administrar los permisos y verificar la configuración. Para registrar un dispositivo e implementar la aplicación de muestra, necesita los siguientes permisos:

- AWSPanoramaFullAccess— Proporciona acceso completo a AWS Panorama, a los puntos de acceso de AWS Panorama en Amazon S3, a las credenciales de los dispositivos y a los registros de los dispositivos en Amazon CloudWatch. AWS Secrets Manager Incluye permiso para crear un rol vinculado a un servicio para AWS Panorama.
- AWS Identity and Access Management (IAM): en la primera ejecución, para crear las funciones utilizadas por el servicio AWS Panorama y el dispositivo AWS Panorama.

Si no tiene permiso para crear roles en IAM, pida a un administrador que abra la consola de AWS Panorama y acepte la solicitud de crear roles de servicio.

Registrar y configurar el dispositivo de AWS Panorama

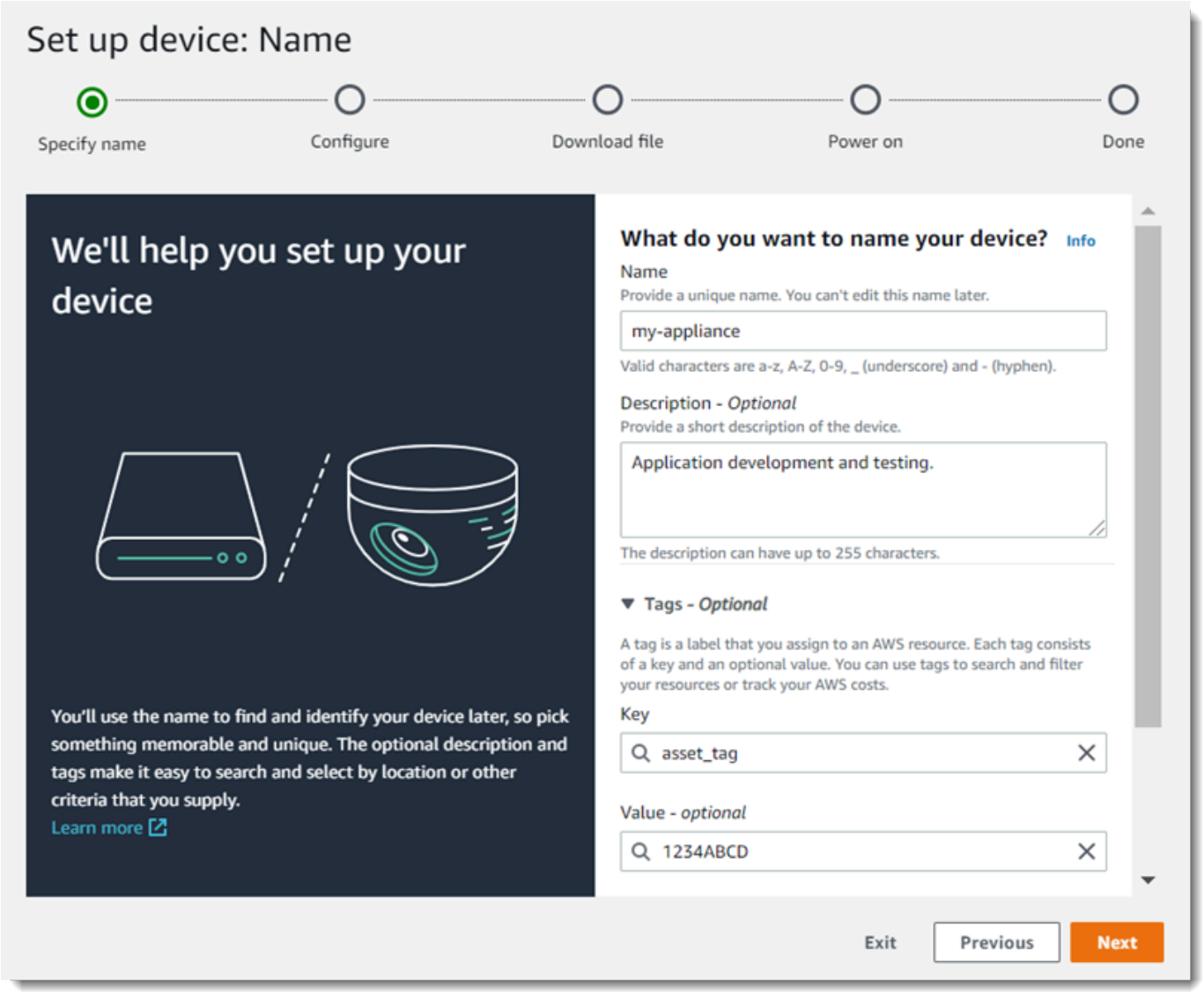
El dispositivo de AWS Panorama es un dispositivo de hardware que se conecta a cámaras habilitadas para la red a través de una conexión de red local. Utiliza un sistema operativo basado en Linux que incluye el SDK de aplicaciones de AWS Panorama y software complementario para ejecutar aplicaciones de visión artificial.

Para conectarse a la AWS administración del dispositivo y la implementación de la aplicación, el dispositivo utiliza un certificado de dispositivo. Utilice la consola de AWS Panorama para generar un certificado de aprovisionamiento. El dispositivo utiliza este certificado temporal para completar la configuración inicial y descargar un certificado de dispositivo permanente.

> **Important**
>
> El certificado de aprovisionamiento que se genera en este procedimiento solo es válido durante 5 minutos. Si no completa el proceso de registro dentro de este plazo, debe volver a empezar.

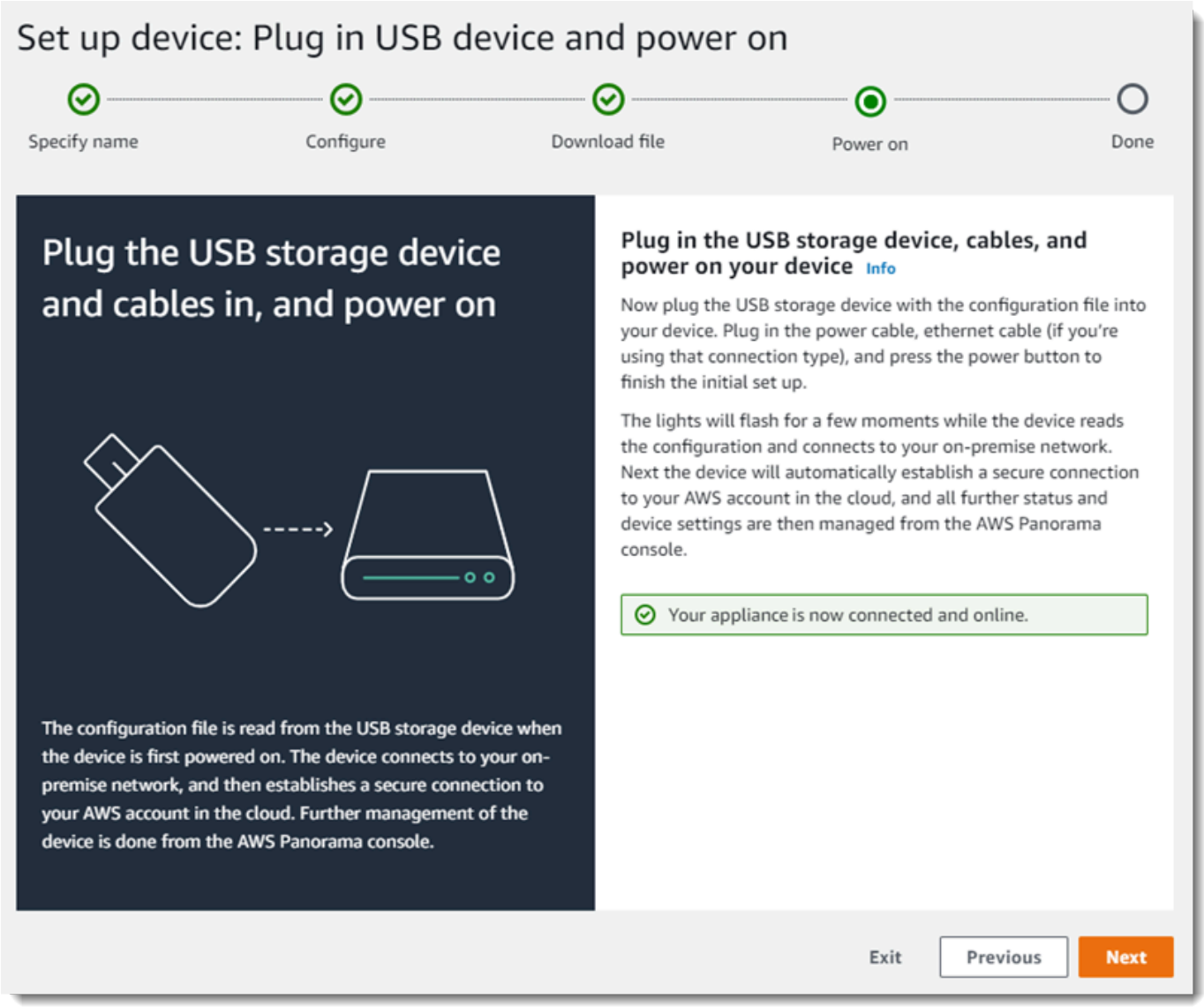
Para registrar un dispositivo

1. Conecte la unidad USB a su equipo. Prepare el dispositivo conectando los cables de red y de alimentación. El dispositivo se enciende y espera a que se conecte una unidad USB.
2. Abra la página Introducción de la consola de AWS Panorama.
3. Elija Añadir dispositivo.
4. Elija Comenzar la configuración.
5. Escriba un nombre y una descripción para el recurso del dispositivo que representa el dispositivo en AWS Panorama. Elija Siguiente.



6. Si necesita asignar manualmente una dirección IP, un servidor NTP o una configuración de DNS, elija Configuración de red avanzada. En caso contrario, elija Siguiente.
7. Elija Descargar archivo. Elija Next (Siguiente).
8. Copie el archivo de configuración en el directorio raíz de la unidad USB.
9. Conecte la unidad USB al puerto USB 3.0 de la parte frontal del dispositivo, junto al puerto HDMI.

   Al conectar la unidad USB, el dispositivo copia el archivo de configuración y el archivo de configuración de red en sí mismo y se conecta a la AWS nube. La luz de estado del dispositivo cambia de verde a azul mientras se completa la conexión y, a continuación, vuelve a ponerse verde.
10. Para continuar, elija Siguiente.



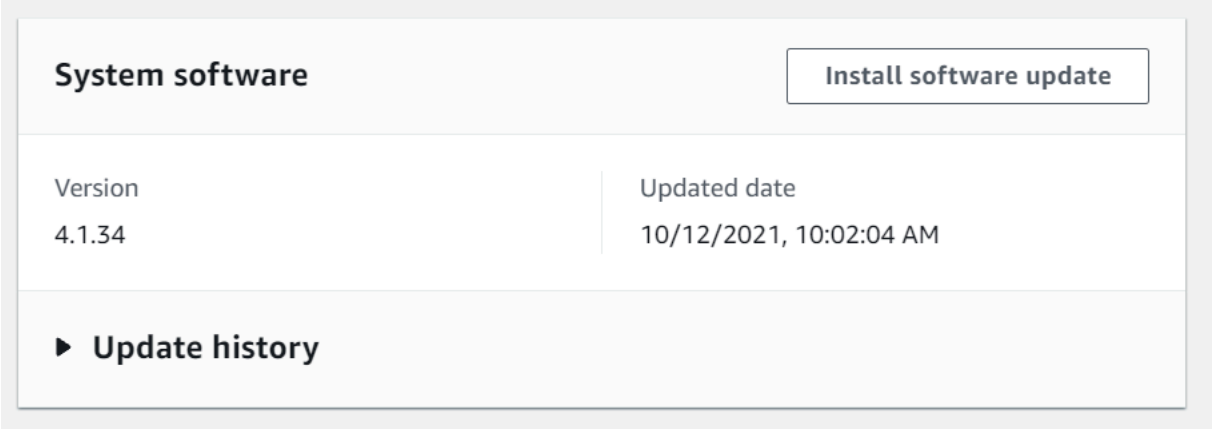
11. Seleccione Listo.

Actualizar el software del dispositivo

El dispositivo de AWS Panorama tiene varios componentes de software, como un sistema operativo Linux, el SDK de aplicaciones de AWS Panorama y bibliotecas y marcos complementarios de visión artificial. Para asegurarse de poder utilizar las características y aplicaciones más recientes con su dispositivo, actualice su software después de la configuración y siempre que haya una actualización disponible.

Para actualizar el software del dispositivo

1. Abra la página Dispositivos de la consola de AWS Panorama.
2. Elija un dispositivo.
3. Elija Configuraciones
4. Elija Instalar actualización de software en Software del sistema.



5. Elija una nueva versión y, a continuación, seleccione Instalar.

> **Important**
>
> Antes de continuar, extraiga la unidad USB del dispositivo y formatéela para eliminar su contenido. El archivo de configuración contiene datos confidenciales y no se elimina automáticamente.

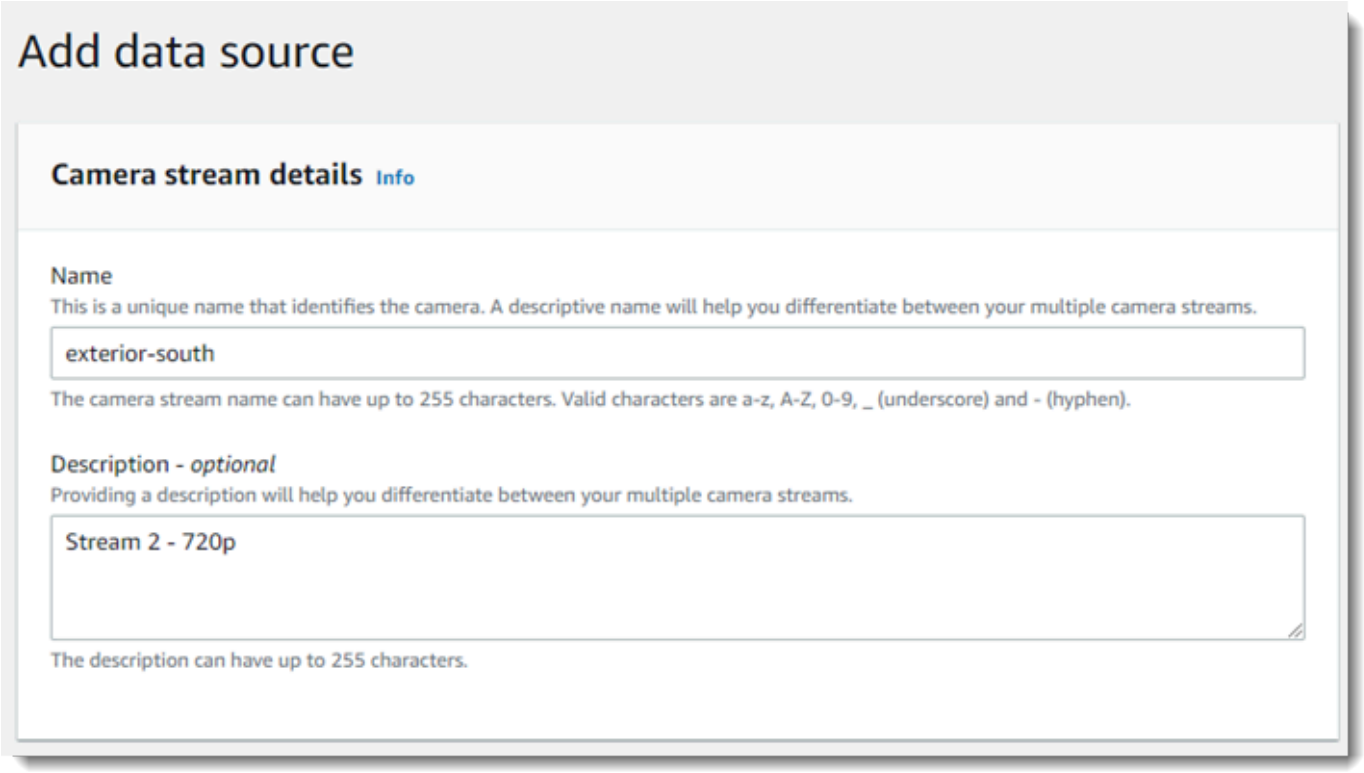
El proceso de actualización puede tardar 30 minutos o más. Puede supervisar su progreso en la consola de AWS Panorama o en un monitor conectado. Cuando se complete el proceso, el dispositivo se reiniciará.

Añadir una transmisión de cámara

A continuación, registre una transmisión de cámara con la consola de AWS Panorama.

Para registrar una transmisión de cámara

1. Abra la página Orígenes de datos de la consola de AWS Panorama.
2. Elija Agregar origen de datos.



3. Configure los siguientes ajustes.

   - Nombre: un nombre para la transmisión de la cámara.
   - Descripción: una breve descripción de la cámara, su ubicación u otros detalles.
   - RTSP URL: una URL que especifica la dirección IP de la cámara y la ruta a la transmisión. Por ejemplo, `rtsp://192.168.0.77/live/mpeg4/` .
   - Credenciales: si la transmisión de la cámara está protegida con contraseña, especifique el nombre de usuario y la contraseña.
4. Seleccione Guardar.

AWS Panorama almacena las credenciales de la cámara de forma segura en AWS Secrets Manager. Varias aplicaciones pueden procesar la misma transmisión de cámara simultáneamente.

Pasos a seguir a continuación

Si ha detectado errores durante la configuración, consulte Solución de problemas.

Para implementar una aplicación de ejemplo, continúe con el tema siguiente.

Implementación de la aplicación de muestra de AWS Panorama

Tras configurar su dispositivo de AWS Panorama o dispositivo compatible y actualizar su software, implemente una aplicación de muestra. En las siguientes secciones, importará una aplicación de muestra con la CLI de la aplicación de AWS Panorama y la implementará con la consola de AWS Panorama.

La aplicación de muestra utiliza un modelo de machine learning para clasificar los objetos en fotogramas de vídeo de una cámara de red. Utiliza el SDK de la aplicación de AWS Panorama para cargar un modelo, obtener imágenes y ejecutar el modelo. A continuación, la aplicación superpone los resultados sobre el vídeo original y los envía a una pantalla conectada.

En un entorno minorista, el análisis de los patrones de tráfico peatonal permite predecir los niveles de tráfico. Al combinar el análisis con otros datos, puede planificar el aumento de las necesidades de personal durante las fiestas y otros eventos, medir la eficacia de los anuncios y las promociones de ventas u optimizar la ubicación de los expositores y la gestión del inventario.

Secciones

Requisitos previos

Para seguir los procedimientos de este tutorial, necesitará un shell o un terminal de línea de comando para ejecutar los comandos. En las listas de código, los comandos van precedidos del símbolo del sistema ($) y del nombre del directorio actual, si es aplicable.

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

Para comandos largos, utilizamos un carácter de escape (\) para dividir un comando en varias líneas.

En Linux y macOS, use su administrador de intérprete de comandos y paquetes preferido. En Windows 10, puede instalar Windows Subsystem para Linux para obtener una versión de Ubuntu y Bash integrada con Windows. Si necesita ayuda para configurar un entorno de desarrollo en Windows, consulte Configuración de un entorno de desarrollo en Windows.

Utilice Python para desarrollar aplicaciones de AWS Panorama e instalar herramientas con pip, el administrador de paquetes de Python. Si aún no dispone de Python, instale la última versión. Si tiene Python 3 pero no pip, instale pip con el administrador de paquetes de su sistema operativo o instale una nueva versión de Python, que contenga pip.

En este tutorial, utilizará Docker para crear el contenedor en el que se ejecuta el código de su aplicación. Instale Docker desde el sitio web de Docker: Obtener Docker

Este tutorial utiliza la CLI de la aplicación de AWS Panorama para importar la aplicación de muestra, crear paquetes y cargar artefactos. La CLI de la aplicación AWS Panorama usa AWS Command Line Interface (AWS CLI) para llamar a las operaciones de la API del servicio. Si ya la tiene AWS CLI, actualícela a la versión más reciente. Para instalar la CLI de la aplicación AWS Panorama y AWS CLI, utilice`pip`.

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

Descargue la aplicación de muestra y extráigala en su espacio de trabajo.

- Ejemplo de aplicación: aws-panorama-sample.zip

Importe la aplicación de ejemplo

Para importar la aplicación de muestra para usarla en su cuenta, utilice la CLI de la aplicación de AWS Panorama. Las carpetas y el manifiesto de la aplicación contienen referencias a un marcador de posición de número de cuenta. Para actualizarlos con su número de cuenta, ejecute el comando `panorama-cli import-application`.

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

El paquete `SAMPLE_CODE`, en el directorio `packages`, contiene el código y la configuración de la aplicación, incluido un Dockerfile que utiliza la imagen base de la aplicación, `panorama-application`. Para crear el contenedor de aplicaciones que se ejecuta en el dispositivo, utilice el 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
```

El último paso con la CLI de la aplicación de AWS Panorama consiste en registrar el código y los nodos del modelo de la aplicación y cargar los activos en un punto de acceso de Amazon S3 proporcionado por el servicio. Los activos incluyen la imagen del contenedor del código, el modelo y un archivo descriptor para cada uno. Para registrar los nodos y cargar los activos, ejecute el comando `panorama-cli package-application`.

```
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
```

Implemente de la aplicación

Utilice la consola de AWS Panorama para implementar la aplicación en su dispositivo.

Para implementar la aplicación de

1. Abra la página Aplicaciones implementadas de la consola de AWS Panorama.
2. Elija Implementar aplicación.
3. Pegue el contenido del manifiesto de la aplicación, `graphs/aws-panorama-sample/graph.json`, en el editor de texto. Elija Siguiente.
4. En Nombre de la aplicación, escriba `aws-panorama-sample`.
5. Elija Proceder a implementar.
6. Elija Comenzar la implementación.
7. Elija Siguiente sin seleccionar un rol.
8. Elija Seleccionar dispositivo y, a continuación, elija su dispositivo. Elija Next (Siguiente).
9. En el paso Seleccionar fuentes de datos, elija Ver entradas y añada la transmisión de la cámara como origen de datos. Elija Next (Siguiente).
10. En el paso Configurar, seleccione Siguiente.

11. Elija Implementación y a continuación elija Listo.
12. En la lista de aplicaciones implementadas, elija aws-panorama-sample.

Actualice esta página para ver las actualizaciones o utilice el siguiente script para supervisar la implementación desde la línea de comandos.

Example monitor-deployment.sh

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

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

```
]
...
```

Si la aplicación no comienza a ejecutarse, comprueba los registros de la aplicación y del dispositivo en Amazon CloudWatch Logs.

Ver resultado

Cuando se completa la implementación, la aplicación comienza a procesar la transmisión de vídeo y envía los registros a CloudWatch.

Para ver los registros en CloudWatch Logs

1. Abra la página de grupos de registros de la consola de CloudWatch registros.
2. Encuentre los registros de aplicaciones y dispositivos de AWS Panorama en los siguientes grupos:

   - Registros de dispositivos: `/aws/panorama/devices/`*`device-id`*
   - Registros de aplicaciones: `/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
```

Para ver la salida de vídeo de la aplicación, conecte el dispositivo a un monitor con un cable HDMI. De forma predeterminada, la aplicación muestra cualquier resultado de clasificación que tenga más del 20% de confianza.

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",
...
```

El modelo de muestra tiene 1000 clases que incluyen muchos animales, alimentos y objetos comunes. Intente apuntar la cámara hacia un teclado o una taza de café.

Para simplificar, la aplicación de muestra utiliza un modelo ligero de clasificación. El modelo genera una matriz única con una probabilidad para cada una de sus clases. Las aplicaciones del mundo real utilizan con mayor frecuencia modelos de detección de objetos que tienen una salida multidimensional. Para ver ejemplos de aplicaciones con modelos más complejos, consulte Ejemplos de aplicaciones, scripts y plantillas.

Activar el SDK de Python

La aplicación de ejemplo la utiliza AWS SDK for Python (Boto) para enviar las métricas a Amazon CloudWatch. Para habilitar esta funcionalidad, cree un rol que conceda permiso a la aplicación para enviar métricas y vuelva a implementar la aplicación con el rol asociado.

La aplicación de ejemplo incluye una AWS CloudFormation plantilla que crea un rol con los permisos que necesita. Para crear el rol, utilice el comando `aws cloudformation deploy`.

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

Para volver a implementar la aplicación

1. Abra la página Aplicaciones implementadas de la consola de AWS Panorama.
2. Elija una aplicación.
3. Elija Reemplazar.
4. Complete los pasos para implementar la aplicación. En Especificar rol de IAM, elija el rol que creó. Su nombre comienza por `aws-panorama-sample-runtime`.
5. Cuando se complete la implementación, abra la CloudWatchconsola y consulte las métricas en el espacio de `AWSPanoramaApplication` nombres. Cada 150 fotogramas, la aplicación registra y carga métricas para el procesamiento de los marcos y el tiempo de inferencia.

Limpieza

Si ha terminado de trabajar con la aplicación de muestra, puede utilizar la consola de AWS Panorama para eliminarla del dispositivo.

Para eliminar la aplicación del dispositivo

1. Abra la página Aplicaciones implementadas de la consola de AWS Panorama.

2. Elija una aplicación.
3. Elija Eliminar del dispositivo.

Pasos a seguir a continuación

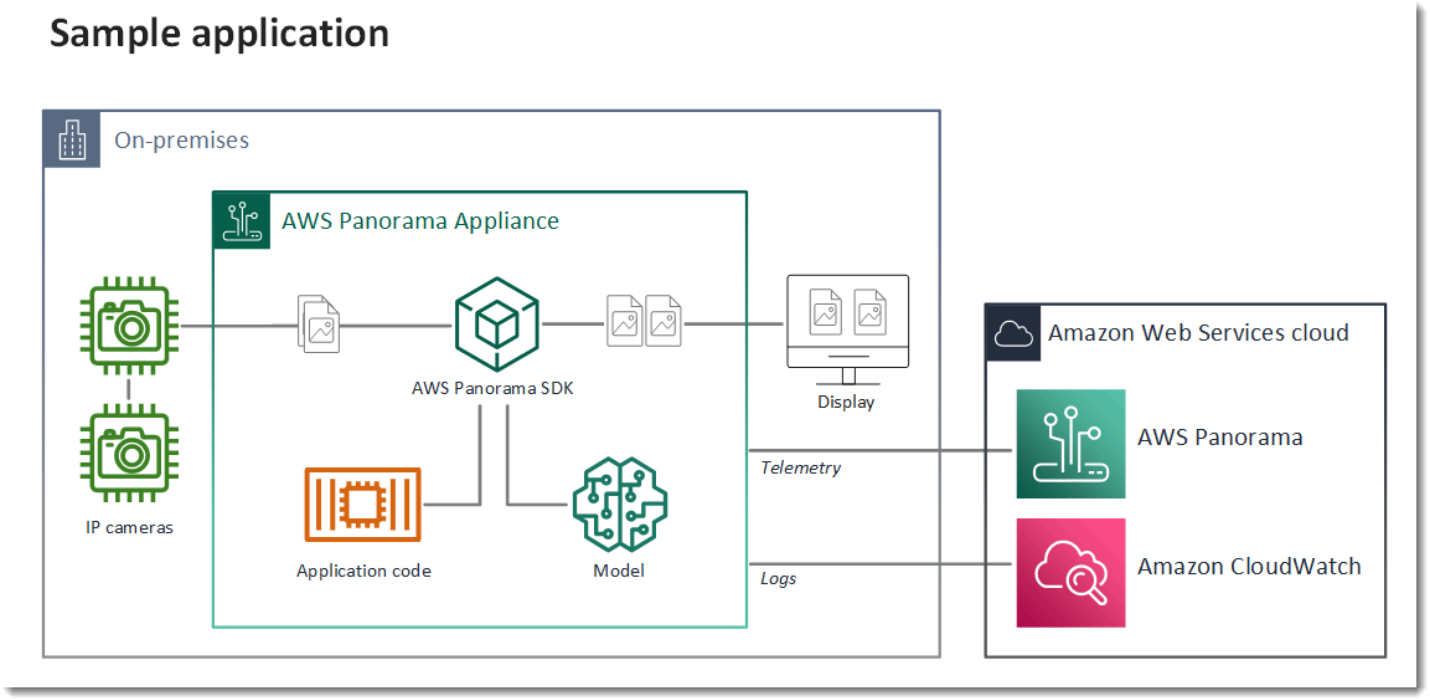
Si ha detectado errores al implementar o ejecutar la aplicación de muestra, consulte Solución de problemas.

Para obtener más información sobre las funciones y la implementación de la aplicación de muestra, continúe con el tema siguiente.

Desarrollo de aplicaciones de AWS Panorama

Puede utilizar la aplicación de muestra de para obtener información sobre la estructura de la aplicación de AWS Panorama y como punto de partida para su propia aplicación.

El siguiente diagrama muestra los componentes principales de la aplicación que se ejecuta en un dispositivo de AWS Panorama. El código de la aplicación utiliza el SDK de aplicaciones de AWS Panorama para obtener imágenes e interactuar con el modelo, al que no tiene acceso directo. La aplicación envía vídeo a una pantalla conectada, pero no envía datos de imagen fuera de la red local.



En este ejemplo, la aplicación utiliza el SDK de aplicaciones de AWS Panorama para obtener fotogramas de vídeo de una cámara, preprocesar los datos de vídeo y enviar los datos a un modelo de visión artificial que detecta objetos. La aplicación muestra el resultado en una pantalla HDMI conectada al dispositivo.

Secciones

- Pasos a seguir a continuación

El manifiesto de la aplicación

El manifiesto de la aplicación es un archivo cuyo nombre es `graph.json` en la carpeta `graphs`. El manifiesto define los componentes de la aplicación, que son paquetes, nodos y periferias.

Los paquetes son archivos de código, configuración y binarios para el código, los modelos, las cámaras y las pantallas de la aplicación. La aplicación de muestra utiliza 4 paquetes:

Example **`graphs/aws-panorama-sample/graph.json`**: paquetes

```
    "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"
        }
    ],
```

Los dos primeros paquetes se definen dentro de la aplicación, en el directorio `packages`. Contienen el código y el modelo específicos de esta aplicación. Los dos segundos paquetes son paquetes genéricos de cámara y pantalla proporcionados por el servicio AWS Panorama. El paquete `abstract_rtsp_media_source` es un marcador de posición para una cámara que usted puede anular durante la implementación. El paquete `hdmi_data_sink` representa el conector de salida HDMI del dispositivo.

Los nodos son interfaces de paquetes, así como parámetros ajenos al paquete que pueden tener valores predeterminados que se anulan en el momento de la implementación. Los paquetes de

código y modelo definen las interfaces en los archivos `package.json` que especifican las entradas y las salidas, que pueden ser transmisiones de vídeo o un tipo de datos básico, como un float, un booleano o una cadena.

Por ejemplo, el nodo `code_node` hace referencia a una interfaz del paquete `SAMPLE_CODE`.

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

Esta interfaz se define en el archivo de configuración del paquete, `package.json`. La interfaz especifica que el paquete es de lógica empresarial y que toma como entradas una secuencia de vídeo denominada `video_in` y un número de coma flotante denominado `threshold`. La interfaz también especifica que el código requiere un búfer de flujo de vídeo denominado `video_out` para enviar el vídeo a una pantalla

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"
                    }
                ]
            }
        ]
    }
}
```

Volviendo al manifiesto de la aplicación, el nodo `camera_node` representa una transmisión de vídeo de una cámara. Incluye un decorador que aparece en la consola cuando despliegue la aplicación y le pide que seleccione una secuencia de cámara.

Example **`graphs/aws-panorama-sample/graph.json`** – Nodo de cámara

```
            {
                "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 de parámetros, `threshold_param`, define el parámetro de umbral de confianza utilizado por el código de la aplicación. Tiene un valor predeterminado de 60 y se puede anular durante la implementación.

Example **`graphs/aws-panorama-sample/graph.json`**: nodo de parámetros

```
            {
                "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 sección final del manifiesto de la aplicación, `edges`, establece conexiones entre nodos. El flujo de vídeo de la cámara y el parámetro de umbral se conectan a la entrada del nodo de código, y la salida de vídeo del nodo de código se conecta a la pantalla.

Example **`graphs/aws-panorama-sample/graph.json`**: periferias

```
        "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"
            }
        ]
```

Crear con la aplicación de muestra

Puede utilizar la aplicación de muestra como punto de partida para su propia aplicación.

El nombre de cada paquete debe ser exclusivo de su cuenta. Si tanto usted como otro usuario de su cuenta utilizan un nombre de paquete genérico, por ejemplo `code` o `model`, es posible que obtengan una versión incorrecta del paquete cuando lo desplieguen. Cambie el nombre del paquete de códigos por uno que represente su aplicación.

Para cambiar el nombre del paquete de códigos

1. Cambie el nombre de la carpeta del paquete: `packages/123456789012-SAMPLE_CODE-1.0/`.

2. Actualice el nombre del paquete en las siguientes ubicaciones.

   - Manifiesto de la aplicación – `graphs/aws-panorama-sample/graph.json`
   - Configuración de paquete – `packages/123456789012-SAMPLE_CODE-1.0/package.json`
   - Script de compilación – `3-build-container.sh`

Para actualizar el código de la aplicación

1. Modifique el código de la aplicación en `packages/123456789012-SAMPLE_CODE-1.0/src/application.py`.
2. Para crear el contenedor, ejecute `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 automáticamente el activo contenedor anterior de la carpeta `assets` y actualiza la configuración del paquete.

3. Para cargar los paquetes, ejecute `4-package-application.py`.
4. Abra la página Aplicaciones implementadas de la consola de AWS Panorama.
5. Elija una aplicación.
6. Elija Reemplazar.
7. Complete los pasos para implementar la aplicación. Si es necesario, puede realizar cambios en el manifiesto de la aplicación, las transmisiones de la cámara o los parámetros.

Cambiar el modelo de visión artificial

La aplicación de muestra incluye un modelo de visión artificial. Para usar su propio modelo, modifique la configuración del nodo del modelo y utilice la CLI de la aplicación de AWS Panorama para importarlo como un activo.

El siguiente ejemplo usa un modelo MXNet SSD ResNet 5.0 que puedes descargar del GitHub repositorio de esta guía: ssd_512_resnet50_v1_voc.tar.gz

Para cambiar el modelo de la aplicación de muestra

1. Cambie el nombre de la carpeta del paquete para que coincida con su modelo. Por ejemplo, para `packages/`*`123456789012`*`-`*`SSD_512_RESNET50_V1_VOC`*`-1.0/`.
2. Actualice el nombre del paquete en las siguientes ubicaciones.

   - Manifiesto de la aplicación – `graphs/aws-panorama-sample/graph.json`
   - Configuración de paquete – `packages/`*`123456789012`*`-`*`SSD_512_RESNET50_V1_VOC`*`-1.0/package.json`
3. En el archivo de configuración del paquete (`package.json`). Cambie el `assets` valor a una matriz en blanco.

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

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

4. Abra el archivo descriptor del paquete (`descriptor.json`). Actualice los valores `framework` y `shape` para que coincidan con su modelo.

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

   El valor de forma, `1,3,512,512`, indica el número de imágenes que el modelo toma como entrada (1), el número de canales de cada imagen (3: rojo, verde y azul) y las dimensiones de la imagen (512 × 512). Los valores y el orden de la matriz varían de un modelo a otro.

5. Importe el modelo con la CLI de la aplicación de AWS Panorama. La CLI de la aplicación de AWS Panorama copia los archivos del modelo y del descriptor en la carpeta `assets` con nombres exclusivos y actualiza la configuración del paquete.

```
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. Para cargar el modelo, ejecute `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. Actualice el código de la aplicación. La mayor parte del código se puede reutilizar. El código específico de la respuesta del modelo está en el método `process_results`.

```
    def process_results(self, inference_results, stream):
        """Processes output tensors from a computer vision model and annotates a
 video frame."""
        for class_tuple in inference_results:
            indexes = self.topk(class_tuple[0])
        for j in range(2):
```

```
            label = 'Class [%s], with probability %.3f.'%
 (self.classes[indexes[j]], class_tuple[0][indexes[j]])
            stream.add_label(label, 0.1, 0.25 + 0.1*j)
```

En función del modelo, es posible que también tenga que actualizar el método `preprocess`.

Preprocesamiento de imágenes

Antes de enviar una imagen al modelo, la prepara para la inferencia redimensionándola y normalizando los datos de color. El modelo que utiliza la aplicación requiere una imagen de 224 × 224 píxeles con tres canales de color, para que coincida con el número de entradas de su primera capa. La aplicación ajusta cada valor de color convirtiéndolo en un número entre 0 y 1, restando el valor promedio de ese color y dividiéndolo por la desviación estándar. Por último, combina los canales de color y los convierte en una NumPy matriz que el modelo puede procesar.

Example application.py: preprocesamiento

```
    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)
```

Este proceso proporciona al modelo valores en un rango predecible centrado alrededor de 0. Coincide con el preprocesamiento aplicado a las imágenes del conjunto de datos de entrenamiento, que es un enfoque estándar, pero puede variar según el modelo.

Carga de métricas con el SDK para Python

La aplicación de ejemplo usa el SDK para Python para cargar métricas en Amazon CloudWatch.

Example application.py: SDK para 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.")
```

Obtiene el permiso de un rol de tiempo de ejecución que usted asigna durante la implementación. La función se define en la `aws-panorama-sample.yml` AWS CloudFormation plantilla.

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/
```

La aplicación de muestra instala el SDK para Python y otras dependencias con pip. Al crear el contenedor de aplicaciones, `Dockerfile` ejecuta comandos para instalar bibliotecas sobre lo que viene con la imagen base.

Example Dockerfile

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

Para usar el AWS SDK en el código de la aplicación, primero modifique la plantilla para añadir permisos para todas las acciones de la API que utilice la aplicación. Actualiza la AWS CloudFormation pila ejecutándola `1-create-role.sh` cada vez que realices un cambio. Luego, implemente los cambios en el código de su aplicación.

En el caso de las acciones que modifican o utilizan los recursos existentes, se recomienda minimizar el alcance de esta política especificando un nombre o patrón para el objetivo `Resource` en una declaración aparte. Para obtener información sobre las acciones y los recursos que admite cada servicio, consulte Acciones, recursos y claves de condiciones en la Referencia de autorización de servicios

Pasos a seguir a continuación

Para obtener instrucciones sobre el uso de la CLI de aplicaciones de AWS Panorama para crear aplicaciones y paquetes desde cero, consulte el README de la CLI.

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

Para obtener más código de muestra y una utilidad de prueba que pueda usar para validar el código de la aplicación antes de la implementación, visite el repositorio de muestras de AWS Panorama.

- github. com/aws-samples/aws-muestras panorámicas

Modelos y cámaras de visión artificial compatibles

AWS Panorama admite modelos creados con PyTorch Apache MXNet y TensorFlow. Cuando implementa una aplicación, AWS Panorama compila su modelo en SageMaker AI Neo. Puede crear modelos en Amazon SageMaker AI o en su entorno de desarrollo, siempre que utilice capas que sean compatibles con SageMaker AI Neo.

Para procesar vídeo y obtener imágenes para enviarlas a un modelo, el dispositivo de AWS Panorama se conecta a una transmisión de vídeo codificada en H.264 mediante el protocolo RTSP. AWS Panorama prueba la compatibilidad de una variedad de cámaras comunes.

Secciones

Modelos compatibles

Cuando crea una aplicación para AWS Panorama, proporciona un modelo de machine learning que la aplicación utiliza para la visión artificial. Puede usar modelos prediseñados y entrenados previamente proporcionados por marcos de modelos, un modelo de muestra o un modelo que cree y entrene usted mismo.

> **Note**
>
> Para ver una lista de los modelos prediseñados que se han probado con AWS Panorama, consulte Compatibilidad del modelo.

AI implementar una aplicación, AWS Panorama utiliza el compilador SageMaker AI Neo para compilar su modelo de visión artificial. SageMaker AI Neo es un compilador que optimiza los modelos para que se ejecuten de manera eficiente en una plataforma de destino, que puede ser una instancia en Amazon Elastic Compute Cloud EC2 (Amazon) o un dispositivo periférico como el AWS Panorama Appliance.

AWS Panorama es compatible con las versiones de PyTorch Apache MXNet y SageMaker AI Neo TensorFlow compatibles con los dispositivos periféricos. Al crear su propio modelo, puede utilizar las versiones del marco que se indican en las notas de la versión de SageMaker AI Neo. En la SageMaker IA, puedes usar el algoritmo de clasificación de imágenes integrado.

Para obtener más información acerca del uso de modelos en AWS Panorama, consulte Modelos de visión artificial.

Cámaras compatibles

El dispositivo de AWS Panorama admite transmisiones de vídeo H.264 desde cámaras que emiten RTSP a través de una red local. Para las transmisiones de cámara de más de 2 megapíxeles, el dispositivo reduce la imagen a 1920 × 1080 píxeles o un tamaño equivalente que conserve la relación de aspecto de la transmisión.

Se ha probado la compatibilidad de los siguientes modelos de cámara con el dispositivo de AWS Panorama:

- Axis: M3057-PLVE, M3058-PLVE, P1448-LE, P3225-LV Mk II
- LaView— LV: 0.40 W PB3
- Vivotek — 0-H IB936
- Amcrest — M-841B IP2
- Anpviz: IPC-B850W-S-3X, IPC-D250W-S
- WGCC: Dome PoE 4MP ONVIF

Para conocer las especificaciones de hardware del dispositivo, consulte Especificaciones del dispositivo de AWS Panorama.

Especificaciones del dispositivo de AWS Panorama

El dispositivo de AWS Panorama tiene las siguientes especificaciones de hardware. Para obtener información sobre otros dispositivos compatibles, consulte la documentación del fabricante.

Componente	Especificación
Procesador y GPU	Nvidia Jetson AGX Xavier con 32 GB de RAM
Ethernet	Dos 1000 Base-T (Gigabyte)
USB	Un USB 2.0 y un USB 3.0 Type-A hembra
Salida HDMI	2.0a
Dimensiones	7,75” × 9,6” × 1,6” (197 mm × 243 mm × 40 mm)
Peso	3,7 lb (1,7 kg)
Fuente de alimentación	100 V-240 V 50-60 Hz CA 65 W
Entrada de alimentación	Receptáculo IEC 60320 C6 (3 pines)
Protección contra polvo y líquidos	IP-62
Conformidad normativa EMI/EMC	FCC Parte 15 (EE. UU.)
Límites de contacto térmico	IEC-62368
Temperatura de funcionamiento	−20 °C a 60 °C
Humedad de funcionamiento	0 % a 95 % de humedad relativa
Temperatura de almacenamiento	−20 °C a 85 °C
Humedad de almacenamiento	No controlado a baja temperatura. 90 % de humedad relativa a alta temperatura
Enfriamiento	Extracción de calor por aire forzado (ventilador)
Opciones de montaje	Montado en bastidor o de forma independiente

Componente	Especificación
Cable de alimentación	6 pies (1,8 metros)
Control de potencia	Pulsador
Restablecer	Interruptor momentáneo
Estado y red LEDs	LED RGB programable de 3 colores

El dispositivo dispone de dispositivos de almacenamiento para tarjetas SD, Bluetooth y Wi-Fi, pero no se pueden utilizar.

El dispositivo de AWS Panorama incluye dos tornillos para montarlo en un rack de servidores. Puede montar dos aparatos side-by-side en un rack de 19 pulgadas.

Service Quotas

AWS Panorama aplica cuotas a los recursos que cree en su cuenta y a las aplicaciones que implemente. Si usa AWS Panorama en varias AWS regiones, las cuotas se aplican por separado a cada región. Las cuotas de AWS Panorama no son ajustables.

Los recursos de AWS Panorama incluyen dispositivos, paquetes de nodos de aplicaciones e instancias de aplicaciones.

- Dispositivos: hasta 50 dispositivos registrados por región.
- Paquetes de nodos: 50 paquetes por región, con hasta 20 versiones por paquete.
- Instancias de aplicaciones: hasta 10 aplicaciones por dispositivo. Cada aplicación puede monitorear hasta 8 transmisiones de cámara. Las implementaciones están limitadas a 200 por día para cada dispositivo.

Cuando utiliza la aplicación AWS Command Line Interface, la CLI o el AWS SDK de AWS Panorama con el servicio AWS Panorama, se aplican cuotas al número de llamadas a la API que realice. Puede realizar hasta 5 solicitudes en total por segundo. Un subconjunto de operaciones de API que crean o modifican recursos aplican un límite adicional de una solicitud por segundo.

Para obtener una lista completa de las cuotas, visite la consola de Service Quotas o consulte los puntos de enlace y las cuotas de AWS Panorama en Referencia general de Amazon Web Services.

AWS Panorama permisos

Puede utilizar AWS Identity and Access Management (IAM) para gestionar el acceso al AWS Panorama servicio y a los recursos, como los dispositivos y las aplicaciones. Para los usuarios de su cuenta que lo utilizan AWS Panorama, usted administra los permisos mediante una política de permisos que puede aplicar a las funciones de IAM. Para administrar los permisos de una aplicación, debe crear un rol y asignarlo a la aplicación.

Para administrar los permisos de los usuarios de tu cuenta, usa la política administrada que se AWS Panorama proporciona o escribe la tuya propia. Necesita permisos para acceder a otros AWS servicios para obtener los registros de aplicaciones y dispositivos, ver las métricas y asignar una función a una aplicación.

Un AWS Panorama dispositivo también tiene una función que le otorga permiso para acceder a AWS los servicios y recursos. La función del dispositivo es una de las funciones de servicio que el AWS Panorama servicio utiliza para acceder a otros servicios en su nombre.

Una función de aplicación es una función de servicio independiente que se crea para una aplicación con el fin de concederle permiso para utilizar AWS los servicios con la AWS SDK for Python (Boto). Para crear un rol de aplicación, necesita privilegios administrativos o la ayuda de un administrador.

Puede restringir los permisos de usuario por el recurso al que afecta una acción y, en algunos casos, por condiciones adicionales. Por ejemplo, puede especificar un patrón para el Nombre de recurso de Amazon (ARN) de una aplicación que requiera que un usuario incluya su nombre de usuario en el nombre de las aplicaciones que cree. Para obtener información sobre los recursos y condiciones que admite cada acción, consulte Acciones, recursos y claves de condiciones para AWS Panorama en la referencia de autorizaciones de servicio.

Para obtener más información, consulte ¿Qué es IAM? en la Guía del usuario de IAM.

Temas

- Políticas de IAM basadas en identidad para AWS Panorama
- Roles de servicio y recursos multiservicios de AWS Panorama
- Concesión de permisos a una aplicación

Políticas de IAM basadas en identidad para AWS Panorama

Para conceder a los usuarios de su cuenta acceso a AWS Panorama, utilice políticas basadas en la identidad en AWS Identity and Access Management (IAM). Aplique políticas basadas en identidad a los roles de IAM que están asociados a un usuario. También puede conceder a los usuarios de otra cuenta permiso para asumir un rol en su cuenta y tener acceso a sus recursos de AWS Panorama.

AWS Panorama proporciona a políticas administradas que otorgan acceso a las acciones de la AWS Panorama y, en algunos casos, acceso a otros servicios utilizados para desarrollar y administrar los recursos de AWS Panorama. AWS Panorama actualiza las políticas gestionadas según sea necesario para garantizar que sus usuarios tengan acceso a las nuevas características cuando se publiquen.

- AWSPanoramaFullAccess— Proporciona acceso completo a AWS Panorama, a los puntos de acceso de AWS Panorama en Amazon S3, a las credenciales de los dispositivos y a los registros de los dispositivos en Amazon CloudWatch. AWS Secrets Manager Incluye permiso para crear un rol vinculado a un servicio para AWS Panorama. Ver política

La política de `AWSPanoramaFullAccess` le permite etiquetar los recursos de AWS Panorama, pero no tiene todos los permisos relacionados con etiquetas que utiliza la consola de AWS Panorama. Para conceder estos permisos, añada la siguiente política.

- ResourceGroupsandTagEditorFullAccess— Ver política

La política de `AWSPanoramaFullAccess` no incluye el permiso para comprar dispositivos desde la consola de AWS Panorama. Para conceder estos permisos, añada la siguiente política.

- ElementalAppliancesSoftwareFullAccess— Ver política

Las políticas administradas conceden permiso a las acciones de la API sin restringir los recursos que un usuario puede modificar. Para conseguir un control más preciso, puede crear sus propias políticas que limiten el ámbito de los permisos de un usuario. Utilice la política de acceso total como punto de partida para sus políticas.

> **Creación de roles de servicio**
>
> La primera vez que utilice la consola de AWS Panorama, necesitará permiso para crear el rol de servicio que utiliza el dispositivo de AWS Panorama. Un rol de servicio da a un servicio permiso para administrar recursos o interactuar con otros servicios. Cree este rol antes de conceder el acceso a sus usuarios.

Para obtener más información sobre los recursos y las condiciones que puede utilizar para limitar el alcance de los permisos de un usuario en AWS Panorama, consulte Acciones, recursos y claves de condición de AWS Panorama en la Referencia de autorización de servicios.

Roles de servicio y recursos multiservicios de AWS Panorama

AWS Panorama usa otros servicios de AWS Panorama para administrar el dispositivo de AWS Panorama, almacenar datos e importar recursos de aplicaciones. Un rol de servicio da a un servicio permiso para administrar recursos o interactuar con otros servicios. Cuando inicia sesión en la consola de AWS Panorama por primera vez, crea los roles de servicio siguientes:

- AWSServiceRoleForAWSPanorama— Permite a AWS Panorama gestionar los recursos en AWS IoT, AWS Secrets Manager y AWS Panorama.

  Política gestionada: AWSPanoramaServiceLinkedRolePolicy
- AWSPanoramaApplianceServiceRole— Permite a un dispositivo AWS Panorama cargar registros y obtener objetos de los puntos de acceso de Amazon S3 creados por AWS Panorama. CloudWatch

  Política gestionada: AWSPanoramaApplianceServiceRolePolicy

Para ver los permisos asociados a cada rol, utilice la consola de IAM. Siempre que sea posible, los permisos del rol se restringen a los recursos que coincidan con un patrón de nomenclatura que utiliza AWS Panorama. Por ejemplo, solo `AWSServiceRoleForAWSPanorama` otorga permiso al servicio para acceder a AWS IoT los recursos que tienen `panorama` en su nombre.

Secciones

Asegurar el rol del dispositivo

El dispositivo de AWS Panorama usa el rol de `AWSPanoramaApplianceServiceRole` para acceder a los recursos de su cuenta. El dispositivo tiene permiso para cargar registros en Logs, leer las credenciales de transmisión de AWS Secrets Manager cámara y acceder a los artefactos de la aplicación en los puntos de acceso de Amazon Simple Storage Service (Amazon S3) que crea AWS Panorama. CloudWatch

> **Note**
>
> Las aplicaciones no utilizan los permisos del dispositivo. Para dar permiso a su aplicación para usar los servicios de AWS , cree un rol de aplicación.

AWS Panorama usa el mismo rol de servicio con todos los dispositivos de su cuenta y no usa roles en todas las cuentas. Para añadir un nivel de seguridad adicional, puede modificar la política de confianza del rol del dispositivo para aplicarlo de forma explícita, lo cual es una práctica recomendada cuando utiliza los roles para conceder a un servicio permiso de acceso a los recursos de su cuenta.

Para actualizar la política de confianza de roles del dispositivo

1. Abra la función del dispositivo en la consola de IAM: AWSPanoramaApplianceServiceRole
2. Elija Editar relación de confianza.
3. Actualice el contenido de la política y, a continuación, seleccione Actualizar política de confianza.

La siguiente política de confianza incluye una condición que garantiza que, cuando AWS Panorama asuma el rol de dispositivo, lo haga para un dispositivo de su cuenta. La condición de `aws:SourceAccount` compara el ID de cuenta especificado por AWS Panorama con el que usted incluye en la política.

Example política de confianza: cuenta específica

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

```
  ]
}
```

Si desea restringir aún más AWS Panorama y permitir que solo asuma el rol con un dispositivo específico, puede especificar el dispositivo por ARN. La condición de `aws:SourceArn` compara el ARN del dispositivo especificado por AWS Panorama con el que usted incluye en la política.

Example política de confianza: dispositivo único

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

Si restablece y vuelve a aprovisionar el dispositivo, debe eliminar temporalmente la condición de ARN de origen y, a continuación, volver a añadirla con el nuevo ID de dispositivo.

Para obtener más información sobre estas condiciones y las prácticas recomendadas de seguridad cuando los servicios utilizan roles para acceder a los recursos de su cuenta, consulte El problema del suplente confuso en la Guía del usuario de IAM.

Utilización de otros servicios

AWS Panorama crea recursos en los siguientes servicios o accede a ellos:

- AWS IoT: cosas, políticas, certificados y trabajos para el dispositivo de AWS Panorama
- Amazon S3: puntos de acceso para organizar modelos, códigos y configuraciones de aplicaciones.
- Secrets Manager: credenciales a corto plazo para el dispositivo de AWS Panorama.

Para obtener información sobre el formato del Nombre de recurso de Amazon (ARN) o los ámbitos de los permisos de cada servicio, consulte los temas de la Guía del usuario de IAM a los que se enlaza en esta lista.

Concesión de permisos a una aplicación

Puede crear un rol para su aplicación a fin de concederle permiso para llamar a AWS los servicios. De forma predeterminada, las aplicaciones no tienen ningún permiso. Debe crear un rol de aplicación en IAM y asignarlo a una aplicación durante la implementación. Para conceder a la aplicación solo los permisos que necesita, cree un rol para ella con permisos para acciones específicas de la API.

La aplicación de ejemplo incluye una AWS CloudFormation plantilla y un script que crean un rol de aplicación. Es un rol de servicio que AWS Panorama puede asumir. Este rol otorga permiso a la aplicación CloudWatch para llamar y cargar métricas.

Example aws-panorama-sample.yml: rol de aplicación

```
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/
```

Puede ampliar este script para conceder permisos a otros servicios especificando una lista de acciones o patrones de la API con el valor de. `Action`

Para obtener más información sobre los permisos en AWS Panorama, consulte AWS Panorama permisos.

Administración del AWS Panorama dispositivo

El AWS Panorama dispositivo es el hardware que ejecuta sus aplicaciones. La AWS Panorama consola se utiliza para registrar un dispositivo, actualizar su software e implementar aplicaciones en él. El software del AWS Panorama dispositivo se conecta a las transmisiones de la cámara, envía fotogramas de vídeo a la aplicación y muestra la salida de vídeo en una pantalla conectada.

Tras configurar su dispositivo u otro dispositivo compatible, debe registrar las cámaras para utilizarlas con las aplicaciones. Las transmisiones de cámara se gestionan en la AWS Panorama consola. Al implementar una aplicación, usted elige qué transmisiones de cámara envía el dispositivo para su procesamiento.

Para ver tutoriales que presentan el AWS Panorama dispositivo con una aplicación de ejemplo, consulteEmpezar con AWS Panorama.

Temas

Administración de un dispositivo de AWS Panorama

Utilice la consola de AWS Panorama para configurar, actualizar o anular el registro del dispositivo de AWS Panorama y otros dispositivos compatibles.

Para configurar un dispositivo, siga las instrucciones del tutorial de introducción. El proceso de configuración crea los recursos en AWS Panorama que rastrean el dispositivo y coordinan las actualizaciones y las implementaciones.

Para registrar un dispositivo con la API de AWS Panorama, consulte Automatizar el registro de dispositivos.

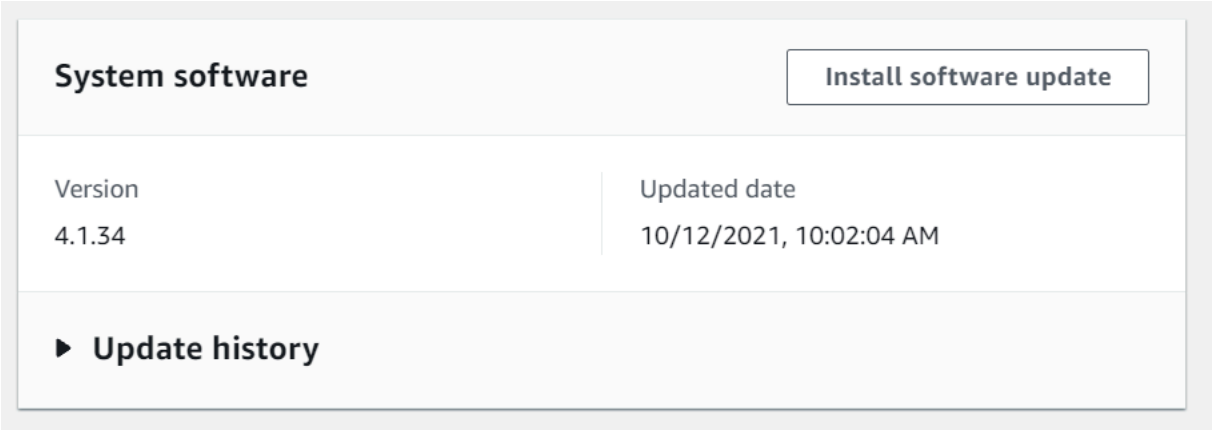
Secciones

Actualice el software del dispositivo

Puede ver e implementar las actualizaciones de software del dispositivo en la consola de AWS Panorama. Las actualizaciones pueden ser obligatorias u opcionales. Cuando haya disponible una actualización obligatoria, la consola le pedirá que la aplique. Puede aplicar las actualizaciones opcionales en la página de Configuración del dispositivo.

Para actualizar el software del dispositivo

1. Abra la página Dispositivos de la consola de AWS Panorama.
2. Elija un dispositivo.
3. Elija Configuraciones
4. Elija Instalar actualización de software en Software del sistema.



5. Elija una nueva versión y, a continuación, seleccione Instalar.

Anulación del registro de un dispositivo

Si ha terminado de trabajar con un dispositivo, puede utilizar la consola de AWS Panorama para anular su registro y eliminar los recursos asociados AWS IoT .

Para eliminar un dispositivo

1. Abra la página Dispositivos de la consola de AWS Panorama.
2. Elija el nombre del dispositivo.
3. Elija Eliminar.
4. Introduzca el nombre del dispositivo y elija Eliminar.

Al eliminar un dispositivo del servicio AWS Panorama, los datos del dispositivo no se eliminan automáticamente. Un dispositivo cuyo registro se ha cancelado no se puede conectar a AWS los servicios y no se puede volver a registrar hasta que se restablezca.

Reinicio de un dispositivo

Puede reiniciar un dispositivo de forma remota.

Para reiniciar un dispositivo

1. Abra la página Dispositivos de la consola de AWS Panorama.
2. Elija el nombre del dispositivo.
3. Elija Reboot.

La consola envía un mensaje al dispositivo para que se reinicie. Para recibir la señal, el dispositivo debe poder conectarse a AWS IoT. Para reiniciar un dispositivo con la API de AWS Panorama, consulte Reinicio de dispositivos.

Restablecer un dispositivo

Para utilizar un dispositivo en una región diferente o con una cuenta de diferente, debe restablecerlo y volver a aprovisionarlo con un nuevo certificado. Al restablecer el dispositivo, se aplica la versión de software requerida más reciente y se eliminan todos los datos de la cuenta.

Para iniciar una operación de restablecimiento, el dispositivo debe estar enchufado y apagado. Mantenga pulsados los botones de encendido y restablecimiento durante cinco segundos. Al soltar los botones, la luz de estado parpadea en color naranja. Espere a que la luz de estado parpadee en color verde antes de aprovisionar o desconectar el dispositivo.

También puede restablecer el software del dispositivo sin eliminar los certificados del dispositivo. Para obtener más información, consulte Botones de encendido y reinicio.

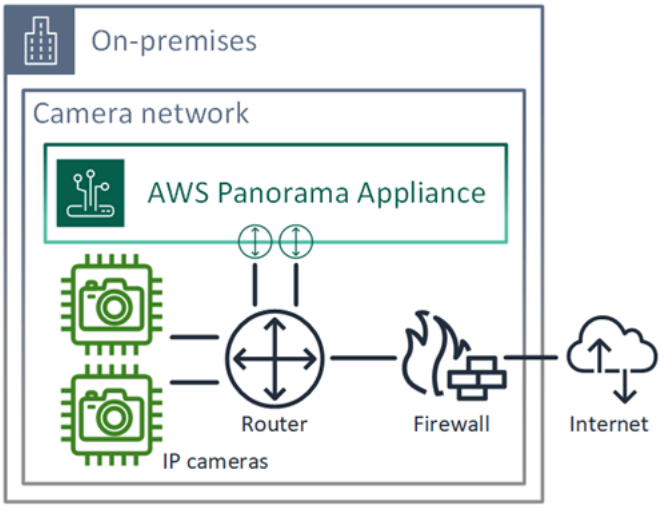
Conexión del dispositivo de AWS Panorama a su red

El dispositivo AWS Panorama requiere conectividad tanto con la AWS nube como con la red local de cámaras IP. Puede conectar el dispositivo a un único firewall que permita el acceso a ambos o conectar cada una de las dos interfaces de red del dispositivo a una subred diferente. En cualquier caso, debe proteger las conexiones de red del dispositivo para evitar el acceso no autorizado a las transmisiones de la cámara.

Secciones

Configuración de red única

El dispositivo tiene dos puertos Ethernet. Si enruta todo el tráfico hacia y desde el dispositivo a través de un único enrutador, puede utilizar el segundo puerto como redundancia en caso de que se interrumpa la conexión física con el primer puerto. Configure el enrutador para permitir que el dispositivo se conecte únicamente a las transmisiones de la cámara y a Internet y para impedir que las transmisiones de la cámara salgan de la red interna.

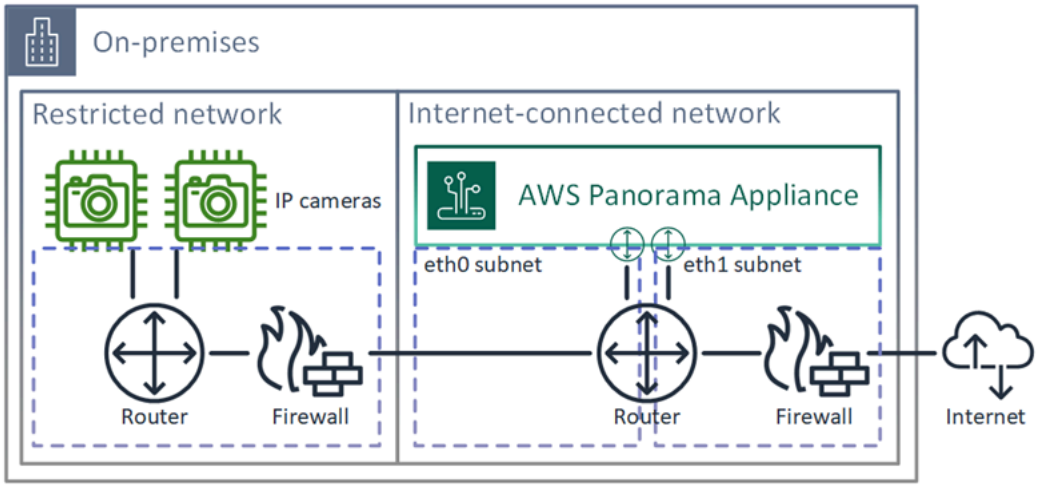


Para obtener información detallada sobre los puertos y puntos de conexión a los que el dispositivo necesita acceder, consulte Configuración del acceso al servicio y Configuración del acceso a la red local.

Configuración de red dual

Para un nivel de seguridad adicional, puede colocar el dispositivo en una red conectada a Internet separada de la red de cámaras. Un firewall entre la red de cámaras restringida y la red del dispositivo solo permite que el dispositivo acceda a las transmisiones de vídeo. Si la red de cámaras estaba previamente aislada por motivos de seguridad, es posible que prefiera este método en lugar de conectar la red de cámaras a un enrutador que también permita el acceso a Internet.

El siguiente ejemplo muestra el dispositivo conectándose a una subred diferente en cada puerto. El enrutador coloca la interfaz `eth0` en una subred que se enruta a la red de cámaras y `eth1` en una subred que se enruta a Internet.



Puede confirmar la dirección IP y la dirección MAC de cada puerto en la consola de AWS Panorama.

Configuración del acceso al servicio

Durante el aprovisionamiento, puede configurar el dispositivo para que solicite una dirección IP específica. Elija una dirección IP con antelación para simplificar la configuración del firewall y garantizar que la dirección del dispositivo no cambie si permanece fuera de línea durante un período prolongado.

El dispositivo utiliza AWS servicios para coordinar las actualizaciones e implementaciones de software. Configure el firewall para permitir que el dispositivo se conecte a estos puntos de conexión.

Acceso a Internet

- AWS IoT (HTTPS y MQTT, puertos 443, 8443 y 8883) y terminales de administración de dispositivos AWS IoT Core . Para obtener más información, consulte los Puntos de conexión y cuotas de AWS IoT Device Management en Referencia general de Amazon Web Services.
- AWS IoT credenciales (HTTPS, puerto 443) y subdominios. `credentials.iot.<region>.amazonaws.com`
- Amazon Elastic Container Registry (HTTPS, puerto 443): `api.ecr.<region>.amazonaws.com`, `dkr.ecr.<region>.amazonaws.com` y subdominios.
- Amazon CloudWatch (HTTPS, puerto 443) —`monitoring.<region>.amazonaws.com`.
- Amazon CloudWatch Logs (HTTPS, puerto 443) —`logs.<region>.amazonaws.com`.
- Amazon Simple Storage Service (HTTPS, puerto 443): `s3.<region>.amazonaws.com`, `s3-accesspoint.<region>.amazonaws.com` y subdominios.

Si su aplicación llama a otros AWS servicios, el dispositivo también necesita acceder a los puntos finales de esos servicios. Para obtener más información, consulte Puntos de conexión y cuotas de servicios.

Configuración del acceso a la red local

El dispositivo necesita acceder a las transmisiones de vídeo RTSP de forma local, pero no a través de Internet. Configure el firewall para permitir que el dispositivo acceda internamente a las transmisiones RTSP en el puerto 554 y no permita que las transmisiones entren o salgan de Internet.

Acceso local

- Protocolo de transmisión en tiempo real (RTSP, puerto 554): para leer las transmisiones de la cámara.
- Protocolo de tiempo de red (NTP, puerto 123): para mantener sincronizado el reloj del dispositivo. Si no ejecuta un servidor NTP en la red, el dispositivo también se puede conectar a servidores NTP públicos a través de Internet.

Conectividad privada

El dispositivo AWS Panorama no necesita acceso a Internet si lo implementa en una subred de VPC privada con una conexión de VPN a. AWS Puede usar una Site-to-Site VPN o AWS Direct Connect

crear una conexión VPN entre un router local y. AWS Dentro de su subred de VPC privada, crea puntos de enlace que permiten que el dispositivo se conecte a Amazon Simple Storage Service y a otros servicios. AWS IoT Para obtener más información, consulte Conexión de un dispositivo a una subred privada.

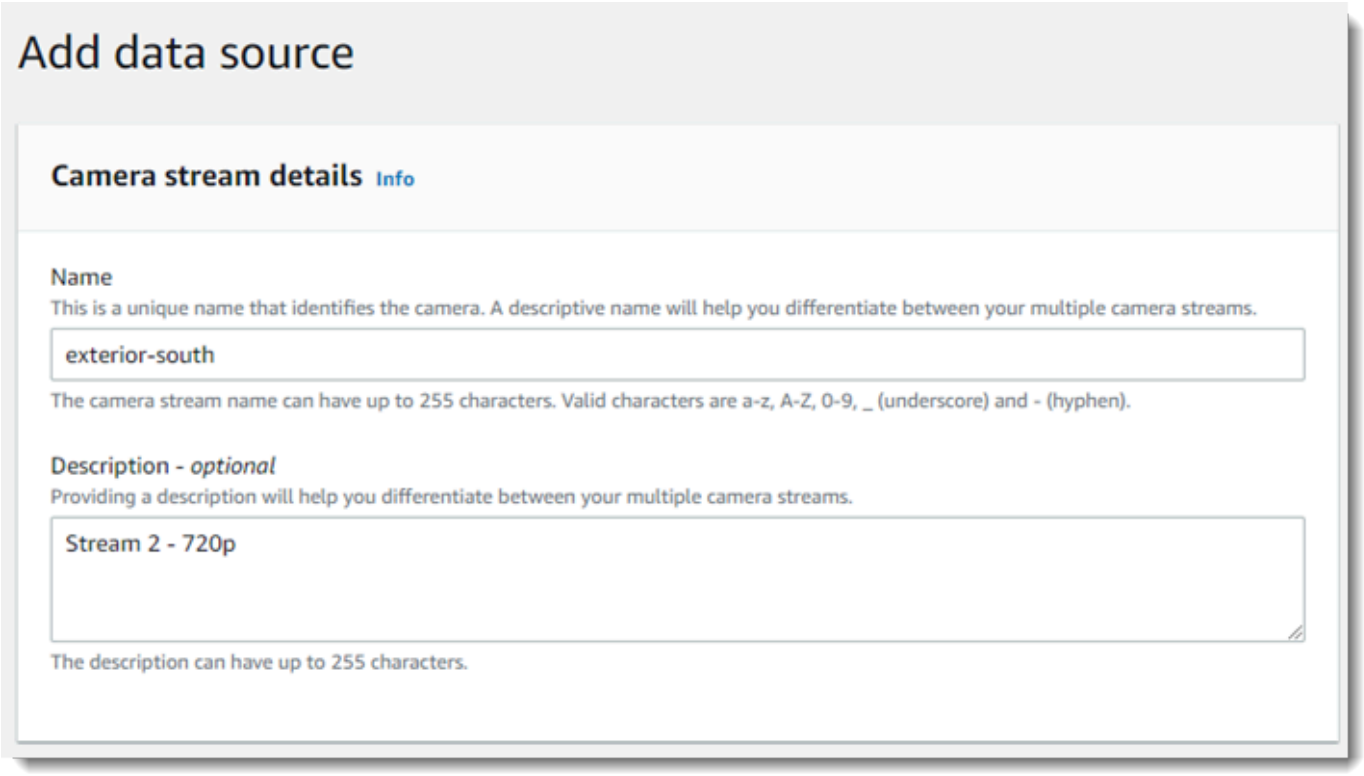
Administración de transmisiones de cámara en AWS Panorama

Para registrar las transmisiones de vídeo como origen de datos para su aplicación, utilice la consola de AWS Panorama. Una aplicación puede procesar varias transmisiones simultáneamente y varios dispositivos pueden conectarse a la misma transmisión.

> **Important**
>
> Una aplicación puede conectarse a cualquier transmisión de cámara que se pueda enrutar desde la red local a la que se conecta. Para proteger las transmisiones de vídeo, configure la red para que solo permita el tráfico RTSP a nivel local. Para obtener más información, consulte Seguridad en AWS Panorama.

Para registrar una transmisión de cámara

1. Abra la página Orígenes de datos de la consola de AWS Panorama.
2. Elija Agregar origen de datos.



3. Configure los siguientes ajustes.

- Nombre: un nombre para la transmisión de la cámara.
- Descripción: una breve descripción de la cámara, su ubicación u otros detalles.
- RTSP URL: una URL que especifica la dirección IP de la cámara y la ruta a la transmisión. Por ejemplo, `rtsp://192.168.0.77/live/mpeg4/` .
- Credenciales: si la transmisión de la cámara está protegida con contraseña, especifique el nombre de usuario y la contraseña.

4. Seleccione Guardar.

Para registrar una transmisión de cámara con la API de AWS Panorama, consulte Automatizar el registro de dispositivos.

Para obtener una lista de cámaras compatibles con el dispositivo de AWS Panorama, consulte Modelos y cámaras de visión artificial compatibles.

Eliminar una transmisión

Puede eliminar una transmisión de cámara en la consola de AWS Panorama.

Para eliminar una transmisión de cámara

1. Abra la página Orígenes de datos de la consola de AWS Panorama.
2. Elija una transmisión de cámara.
3. Elija Borrar origen de datos.

Eliminar una transmisión de cámara del servicio no detiene la ejecución de las aplicaciones ni elimina las credenciales de la cámara de Secrets Manager. Para eliminar secretos, use la consola de Secrets Manager.

Administrar aplicaciones en un dispositivo de AWS Panorama

Una aplicación es una combinación de código, modelos y configuración. Desde la página Dispositivos de la consola de AWS Panorama, puede administrar las aplicaciones del dispositivo.

Para administrar aplicaciones en un dispositivo de AWS Panorama

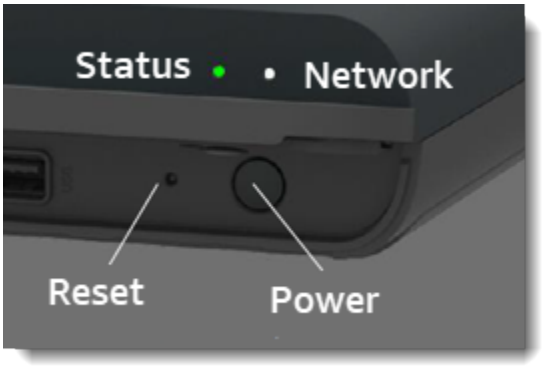
1. Abra la página Dispositivos de la consola de AWS Panorama.
2. Elija un dispositivo.

La página Aplicaciones implementadas muestra las aplicaciones que se han implementado en el dispositivo.

Utilice las opciones de esta página para eliminar las aplicaciones implementadas del dispositivo o reemplazar una aplicación en ejecución por una nueva versión. También puede clonar una aplicación (en ejecución o eliminada) para implementar una nueva copia de la misma.

Botones y luces del dispositivo de AWS Panorama

El dispositivo de AWS Panorama tiene dos luces LED sobre el botón de encendido que indican el estado del dispositivo y la conectividad de la red.



Luz de estado

LEDs Cambian de color y parpadean para indicar el estado. Un parpadeo lento es una vez cada tres segundos. Un parpadeo rápido es una vez por segundo.

Estado: estados del led

- Verde que parpadea rápidamente: el dispositivo se está iniciando.
- Verde fijo: el aparato funciona con normalidad.
- Parpadeo lento en azul: el dispositivo está copiando los archivos de configuración e intentando registrarse en ellos. AWS IoT
- Azul que parpadea rápidamente: el dispositivo está copiando una imagen de registro en una unidad USB.
- Rojo parpadeante rápido: el dispositivo detectó un error durante el inicio o se ha sobrecalentado.
- Naranja parpadeante lento: el dispositivo está restaurando la última versión del software.
- Naranja parpadeante rápido: el dispositivo está restaurando la versión mínima del software.

Luz de red

El led de red tiene los siguientes estados:

Estados del led de red

- Verde fijo: hay un cable Ethernet conectado.

- Verde parpadeante: el dispositivo se está comunicando a través de la red.
- Rojo fijo: no hay ningún cable Ethernet conectado.

Botones de encendido y reinicio

Los botones de encendido y reinicio se encuentran en la parte frontal del dispositivo, debajo de una cubierta protectora. El botón de reinicio es más pequeño y está empotrado. Use un destornillador pequeño o un clip para presionarlo.

Para restablecer un dispositivo

1. El aparato debe estar enchufado y apagado. Para apagar el aparato, mantenga pulsado el botón de encendido durante 1 segundo y espere a que finalice la secuencia de apagado. La secuencia de apagado tarda unos 10 segundos.
2. Para reiniciar el aparato, utilice las siguientes combinaciones de botones. Una pulsación corta dura 1 segundo. Una pulsación larga dura 5 segundos. Para operaciones que requieren varios botones, mantenga pulsados ambos botones simultáneamente.

   - Restablecimiento completo: mantenga presionado el botón de encendido y reinicie.

     Restaura la versión mínima del software y elimina todos los archivos de configuración y las aplicaciones.
   - Restaure la última versión del software: pulse brevemente el botón de reinicio.

     Vuelve a aplicar la última actualización de software al dispositivo.
   - Restaure la versión mínima del software: mantenga pulsada la tecla Restablecer.

     Vuelve a aplicar la última actualización requerida de software al dispositivo.
3. Suelte ambos botones. El aparato se enciende y la luz de estado parpadea en naranja durante varios minutos.
4. Cuando el aparato esté listo, la luz de estado parpadeará en verde.

El restablecimiento de un dispositivo no lo elimina del servicio AWS Panorama. Para obtener más información, consulte Anulación del registro de un dispositivo.

Gestión de AWS Panorama aplicaciones

Las aplicaciones se ejecutan en el AWS Panorama dispositivo para realizar tareas de visión artificial en transmisiones de vídeo. Puede crear aplicaciones de visión artificial combinando código Python y modelos de aprendizaje automático e implementarlas en el AWS Panorama dispositivo a través de Internet. Las aplicaciones pueden enviar vídeo a una pantalla o utilizar el SDK de AWS para enviar los resultados a los servicios de AWS.

Temas

Implementar una aplicación

Para implementar una aplicación, debe utilizar la CLI de la aplicación de AWS Panorama, importarla a su cuenta, crear el contenedor, cargar y registrar activos y crear una instancia de aplicación. En este tema se analiza cada uno de estos pasos en detalle y se describe lo que ocurre en segundo plano.

Si aún no ha implementado una aplicación, consulte Empezar con AWS Panorama para ver un tutorial.

Para obtener más información sobre cómo personalizar y ampliar la aplicación de ejemplo, consulte Creación de AWS Panorama aplicaciones.

Secciones

Para instalar la CLI de la aplicación de AWS Panorama

Para instalar la CLI de la aplicación AWS Panorama y AWS CLI usar pip.

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

Para crear imágenes de aplicaciones con la CLI de la aplicación de AWS Panorama, necesita Docker. En Linux, `qemu` y en sistemas relacionados, también se requieren bibliotecas de sistemas relacionadas. Para obtener más información sobre la instalación y configuración de la CLI de la aplicación AWS Panorama, consulte el archivo README en el GitHub repositorio del proyecto.

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

Para obtener instrucciones sobre cómo configurar un entorno de compilación en Windows con, consulte. WSL2 Configuración de un entorno de desarrollo en Windows

Importar una aplicación

Si está trabajando con una aplicación de muestra o una aplicación proporcionada por un tercero, utilice la CLI de la aplicación de AWS Panorama para importar la aplicación.

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

Este comando cambia el nombre de los paquetes de aplicación con su ID de cuenta. Los nombres de los paquetes comienzan con el ID de cuenta de la cuenta en la que se implementan. Al implementar una aplicación en varias cuentas, debe importar y empaquetar la aplicación por separado para cada cuenta.

Por ejemplo, la aplicación de muestra de esta guía es un paquete de códigos y un paquete modelo, cada uno nombrado con un marcador de posición de identificador de cuenta. El `import-application` comando les cambia el nombre para usar el ID de cuenta que la CLI deduce de las credenciales del espacio de AWS trabajo.

```
/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` se sustituye por su ID de cuenta en los nombres del directorio del paquete y en el manifiesto de la aplicación (`graph.json`), que hace referencia a ellos. Puede confirmar el ID de su cuenta llamando `aws sts get-caller-identity` con el. AWS CLI

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

Crear una imagen de contenedor

El código de la aplicación está empaquetado en una imagen de contenedor de Docker, que incluye el código de la aplicación y las bibliotecas que se instalan en el Dockerfile. Utilice el comando `build-container` de la CLI de la aplicación de AWS Panorama para crear una imagen de Docker y exportar una imagen del sistema de archivos.

```
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
```

Este comando crea una imagen de Docker de nombre `code_asset` y exporta un sistema de archivos a un archivo `.tar.gz` de la carpeta `assets`. La CLI extrae la imagen base de la aplicación desde Amazon Elastic Container Registry (Amazon ECR), tal y como se especifica en el Dockerfile de la aplicación.

Además del archivo contenedor, la CLI crea un activo para el descriptor del paquete (`descriptor.json`). Se cambia el nombre de ambos archivos con un identificador único que refleja un hash del archivo original. La CLI de la aplicación de AWS Panorama también agrega un bloque a

la configuración del paquete que registra los nombres de los dos activos. El dispositivo utiliza estos nombres durante el proceso de implementación.

Example packages/123456789012-SAMPLE_CODE-1.0/package.json – con bloque de activos

```
{
    "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"
                    },
```

El nombre del activo de código, especificado en el comando `build-container`, debe coincidir con el valor del campo `asset` de la configuración del paquete. En el ejemplo anterior, ambos valores son `code_asset`.

Importar un modelo

Es posible que la aplicación tenga un archivo de modelos en la carpeta de activos o que se descargue por separado. Si tiene un modelo nuevo, un modelo actualizado o un archivo descriptor de modelo actualizado, utilice el comando `add-raw-model` para 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
```

Si solo necesita actualizar el archivo descriptor, puede reutilizar el modelo existente en el directorio de activos. Es posible que necesite actualizar el archivo descriptor para configurar funciones como el modo de precisión de punto flotante. Por ejemplo, el siguiente script muestra cómo hacerlo con la aplicación de muestra.

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
```

Los cambios en el archivo descriptor del directorio del paquete del modelo no se aplican hasta que se vuelva a importarlo con la CLI. La CLI actualiza la configuración del paquete del modelo con los nuevos nombres de activos en su lugar, de forma similar a como actualiza la configuración del paquete de códigos de la aplicación cuando se reconstruye un contenedor.

Cargar los activos de la aplicación

Para cargar y registrar los activos de la aplicación, que incluyen el archivo modelo, el archivo del sistema de archivos del contenedor y sus archivos descriptores, utilice el 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
...
```

Si no hay cambios en un archivo de activos o en la configuración del paquete, la CLI los omite.

```
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 carga los activos de cada paquete en un punto de acceso de Amazon S3 específico de su cuenta. AWS Panorama administra el punto de acceso por usted y proporciona información sobre él a través de la DescribePackageAPI. La CLI carga los activos de cada paquete en la ubicación proporcionada para ese paquete y los registra en el servicio AWS Panorama con los ajustes descritos en la configuración del paquete.

Implementar una aplicación con la consola de AWS Panorama

Puede implementar una aplicación con la consola de AWS Panorama. Durante el proceso de implementación, usted elige qué secuencias de cámara desea transferir al código de la aplicación y configura las opciones proporcionadas por el desarrollador de la aplicación.

Para implementar una aplicación

1. Abra la página Aplicaciones implementadas de la consola de AWS Panorama.
2. Elija Implementar aplicación.
3. Pegue el contenido del manifiesto de la aplicación, `graph.json`, en el editor de texto. Elija Next (Siguiente).
4. Introduzca un nombre y una descripción.
5. Elija Proceder a implementar.
6. Elija Comenzar la implementación.
7. Si la aplicación usa un rol, elíjalo en el menú desplegable. Elija Next (Siguiente).
8. Elija Seleccionar dispositivo y, a continuación, elija su dispositivo. Elija Next (Siguiente).
9. En el paso Seleccionar fuentes de datos, elija Ver entradas y añada la transmisión de la cámara como origen de datos. Elija Next (Siguiente).
10. En el paso Configurar, configure los ajustes específicos de la aplicación definidos por el desarrollador. Elija Next (Siguiente).
11. Elija Implementación y a continuación elija Listo.
12. En la lista de aplicaciones implementadas, elija la aplicación para supervisar su estado.

El proceso de implementación tarda entre 15 y 20 minutos. La salida del dispositivo puede permanecer en blanco durante un período prolongado mientras se inicia la aplicación. Si ocurre un error, consulte Solución de problemas.

Automatización de las implementaciones de aplicaciones

Puede automatizar el proceso de implementación de la aplicación con la CreateApplicationInstanceAPI. La API toma dos archivos de configuración como entrada. El manifiesto de la aplicación especifica los paquetes utilizados y sus relaciones. El segundo archivo es un archivo de anulaciones que especifica las anulaciones de los valores del manifiesto de la aplicación en el momento de la implementación. El uso de un archivo de anulaciones permite utilizar el mismo manifiesto de aplicación para implementar la aplicación con diferentes secuencias de cámara y configurar otros ajustes específicos de la aplicación.

Para obtener más información y ejemplos de scripts para cada uno de los pasos de este tema, consulte Automatización de las implementaciones de aplicaciones.

Administración de aplicaciones en la consola de AWS Panorama

Utilice la consola de AWS Panorama para gestionar las aplicaciones implementadas.

Secciones

Actualice o copie una aplicación

Para actualizar una aplicación, utilice la opción Reemplazar. Al reemplazar una aplicación, puede actualizar su código o modelos.

Para actualizar una aplicación

1. Abra la página Aplicaciones implementadas de la consola de AWS Panorama.
2. Elija una aplicación.
3. Elija Reemplazar.
4. Siga las instrucciones para crear una nueva versión o aplicación.

También hay una opción de clonación que actúa de forma similar a Reemplazar, pero no elimina la versión anterior de la aplicación. Puede usar esta opción para probar los cambios en una aplicación sin detener la versión en ejecución o para volver a implementar una versión que ya haya eliminado.

Eliminar versiones y aplicaciones

Para limpiar las versiones de las aplicaciones que no se utilizan, elimínelas de sus dispositivos.

Para eliminar una aplicación de

1. Abra la página Aplicaciones implementadas de la consola de AWS Panorama.
2. Elija una aplicación.
3. Elija Eliminar del dispositivo.

Configuración de paquete

Cuando utiliza el comando `panorama-cli package-application` CLI de la aplicación AWS Panorama, la CLI carga los activos de la aplicación en Amazon S3 y los registra en AWS Panorama. Los activos incluyen archivos binarios (modelos e imágenes de contenedores) y archivos descriptores, que el dispositivo de AWS Panorama descarga durante la implementación. Para registrar los activos de un paquete, debe proporcionar un archivo de configuración de paquete independiente que defina el paquete, sus activos y su interfaz.

El siguiente ejemplo muestra una configuración de paquete para un nodo de código con una entrada y una salida. La entrada de vídeo proporciona acceso a los datos de imagen de una transmisión de cámara. El nodo de salida envía las imágenes procesadas a una pantalla.

Example packages/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 sección `assets` especifica los nombres de los artefactos que la CLI de la aplicación AWS Panorama cargó en Amazon S3. Si importa una aplicación de muestra o una aplicación de otro usuario, esta sección puede estar vacía o hacer referencia a activos que no están en su cuenta. Cuando se ejecuta `panorama-cli package-application`, la CLI de la aplicación AWS Panorama rellena esta sección con los valores correctos.

El manifiesto de la aplicación AWS Panorama

Al implementar una aplicación, proporciona un archivo de configuración denominado “manifiesto de aplicación”. Este archivo define la aplicación como un gráfico con nodos y periferias. El manifiesto de la aplicación forma parte del código fuente de la aplicación y se almacena en el directorio de `graphs`.

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"
            }
        ]
    }
}
```

Los nodos están conectados por periferias, que especifican los mapeos entre las entradas y salidas de los nodos. La salida de un nodo se conecta a la entrada de otro, formando un gráfico.

Esquema JSON

El formato del manifiesto de la aplicación y de los documentos de anulación se define en un esquema JSON. Puede usar el esquema JSON para validar los documentos de configuración antes de la implementación. El esquema JSON está disponible en el GitHub repositorio de esta guía.

- Esquema JSON: aws-panorama-developer-guide/resources

Nodos de aplicación

Los nodos son modelos, códigos, secuencias de cámara, salidas y parámetros. Un nodo tiene una interfaz que define sus entradas y salidas. La interfaz se puede definir en un paquete de su cuenta, en un paquete proporcionado por AWS Panorama o en un tipo integrado.

En el siguiente ejemplo, `code_node` y `model_node` se refieren al código de muestra y los paquetes de modelos incluidos con la aplicación de muestra. `camera_node` utiliza un paquete proporcionado por AWS Panorama para crear un marcador de posición para una transmisión de cámara que usted especifique durante la implementación.

Example graph.json — Nodos

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

Periferias

Las periferias mapean la salida de un nodo a la entrada de otro. En el siguiente ejemplo, la primera periferia mapea la salida de un nodo de transmisión de cámara a la entrada de un nodo de código de aplicación. Los nombres `video_in` y `video_out` se definen en las interfaces de los paquetes de nodos.

Example graph.json — Periferias

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

En el código de tu aplicación, utilizas los atributos `inputs` y `outputs` para obtener imágenes del flujo de entrada y enviarlas al flujo de salida.

Example application.py — Entrada y salida de vídeo

```
    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)
```

Nodos abstractos

En un manifiesto de aplicación, un nodo abstracto hace referencia a un paquete definido por AWS Panorama, que puede utilizar como marcador de posición en el manifiesto de la aplicación. AWS Panorama ofrece dos tipos de nodos abstractos.

- Transmisión de cámara: elija la transmisión de cámara que utilizará la aplicación durante la implementación.

  Nombre del paquete: `panorama::abstract_rtsp_media_source`

  Nombre de la interfaz: `rtsp_v1_interface`

- Salida HDMI: indica que la aplicación emite vídeo.

  Nombre del paquete: `panorama::hdmi_data_sink`

  Nombre de la interfaz: `hdmi0`

El siguiente ejemplo muestra un conjunto básico de paquetes, nodos y periferias para una aplicación que procesa los flujos de cámara y envía vídeo a una pantalla. El nodo de cámara, que utiliza la interfaz del paquete de `abstract_rtsp_media_source` de AWS Panorama, puede aceptar varias secuencias de cámara como entrada. El nodo de salida, al que hace referencia `hdmi_data_sink`, permite que el código de la aplicación acceda a un búfer de vídeo que se emite desde el puerto HDMI del dispositivo.

Example graph.json — Nodos abstractos

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

Parámetros de la aplicación

Los parámetros son nodos que tienen un tipo básico y se pueden anular durante la implementación. Un parámetro puede tener un valor predeterminado y un decorador, que indica al usuario de la aplicación cómo configurarlo.

Tipos de parámetros

- `string` – Una cadena. Por ejemplo, `DEBUG`.
- `int32` – Un número entero. Por ejemplo, `20` .
- `float32` – Un número de coma flotante. Por ejemplo, `47.5` .
- `boolean` – `true` o `false`.

El siguiente ejemplo muestra dos parámetros, una cadena y un número, que se envían a un nodo de código como entradas.

Example graph.json – Parámetros

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

Puedes modificar los parámetros directamente en el manifiesto de la aplicación o proporcionar nuevos valores en el momento de la implementación con sustituciones. Para obtener más información, consulte Configuración en tiempo de implementación con anulaciones.

Configuración en tiempo de implementación con anulaciones

Los parámetros y los nodos abstractos se configuran durante la implementación. Si utiliza la consola de AWS Panorama para la implementación, puede especificar un valor para cada parámetro y elegir una transmisión de cámara como entrada. Si usa la API de AWS Panorama para implementar aplicaciones, debe especificar estos ajustes con un documento de anulaciones.

La estructura de un documento de anulaciones es similar a la de un manifiesto de aplicación. Para los parámetros con tipos básicos, se define un nodo. Para las transmisiones de cámara, se definen un nodo y un paquete que se asignan a una transmisión de cámara registrada. A continuación, defina una anulación para cada nodo que especifique el nodo del manifiesto de la aplicación al que sustituye.

Example overrides.json

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

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

En el ejemplo anterior, el documento define las anulaciones para un parámetro de cadena y un nodo de cámara abstracto. `node0verrides` indica a AWS Panorama qué nodos de este documento anulan cuáles del manifiesto de la aplicación.

Creación de AWS Panorama aplicaciones

Las aplicaciones se ejecutan en el AWS Panorama dispositivo para realizar tareas de visión artificial en transmisiones de vídeo. Puede crear aplicaciones de visión artificial combinando código Python y modelos de aprendizaje automático e implementarlas en el AWS Panorama dispositivo a través de Internet. Las aplicaciones pueden enviar vídeo a una pantalla o utilizar el SDK de AWS para enviar los resultados a los servicios de AWS.

Un modelo analiza las imágenes para detectar personas, vehículos y otros objetos. Basándose en las imágenes que ha visto durante el entrenamiento, el modelo le dice qué cree que es algo y qué tan seguro está al adivinarlo. Puede entrenar modelos con sus propios datos de imagen o empezar con una muestra.

El código de la aplicación procesa imágenes fijas de una secuencia de cámara, las envía a un modelo y procesa el resultado. Un modelo puede detectar varios objetos y devolver sus formas y ubicación. El código puede usar esta información para añadir texto o gráficos al vídeo, o para enviar los resultados a un servicio AWS para su almacenamiento o procesamiento posterior.

Para obtener imágenes de una transmisión, interactuar con un modelo y generar vídeo, el código de la aplicación utiliza el SDK de AWS Panorama aplicaciones. El SDK de la aplicación es una biblioteca de Python que admite modelos generados con PyTorch Apache MXNet y TensorFlow.

Temas

Modelos de visión artificial

Un modelo de visión artificial es un programa de software que está entrenado para detectar objetos en imágenes. Un modelo aprende a reconocer un conjunto de objetos analizando primero las imágenes de esos objetos mediante el entrenamiento. Un modelo de visión artificial toma una imagen como entrada y genera información sobre los objetos que detecta, como el tipo de objeto y su ubicación. AWS Panorama admite modelos de visión artificial creados con PyTorch Apache MXNet y TensorFlow.

> **Note**
>
> Para ver una lista de los modelos prediseñados que se han probado con AWS Panorama, consulte Compatibilidad del modelo.

Secciones

Uso de modelos en código

Un modelo devuelve uno o más resultados, que pueden incluir probabilidades de las clases detectadas, información de ubicación y otros datos. En el siguiente ejemplo, se muestra cómo realizar inferencias en una imagen a partir de una transmisión de vídeo y enviar la salida del modelo a una función de procesamiento.

Example application.py: inferencia

```
    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)
```

El siguiente ejemplo muestra una función que procesa los resultados del modelo de clasificación básico. El modelo de muestra devuelve una matriz de probabilidades, que es el primer y único valor de la matriz de resultados.

Example application.py: procesando los resultados

```
   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)
```

El código de la aplicación busca los valores con las probabilidades más altas y los asigna a las etiquetas de un archivo de recursos que se carga durante la inicialización.

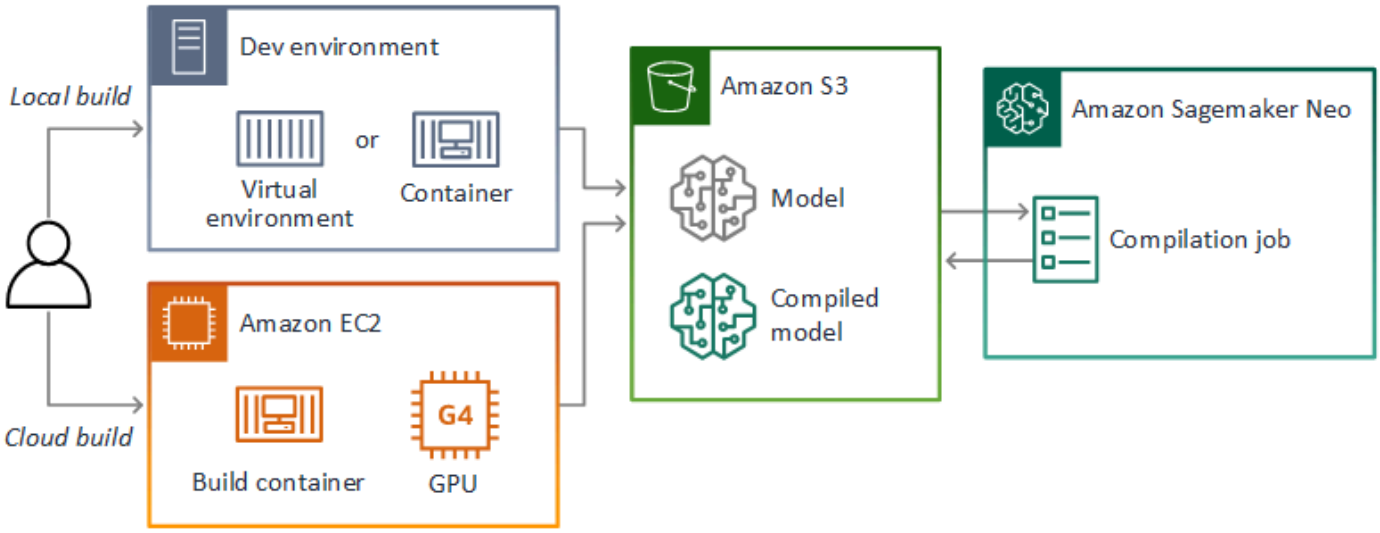
Creación de un modelo personalizado

Puede usar los modelos que haya creado PyTorch, en Apache MXNet y TensorFlow en las aplicaciones de AWS Panorama. Como alternativa a la creación y el entrenamiento de modelos en SageMaker IA, puedes usar un modelo entrenado o crear y entrenar tu propio modelo con un marco compatible y exportarlo a un entorno local o a Amazon EC2.

> **Note**
>
> Para obtener más información sobre las versiones de los marcos y los formatos de archivo compatibles con SageMaker AI Neo, consulte los marcos compatibles en la Guía para desarrolladores de Amazon SageMaker AI.

El repositorio de esta guía proporciona un ejemplo de aplicación que muestra este flujo de trabajo para un modelo de Keras en TensorFlow `SavedModel` formato. Utiliza TensorFlow 2 y se puede ejecutar localmente en un entorno virtual o en un contenedor Docker. La aplicación de muestra también incluye plantillas y scripts para crear el modelo en una EC2 instancia de Amazon.

- Ejemplo de aplicación de modelo personalizado



AWS Panorama usa SageMaker AI Neo para compilar modelos para usarlos en el dispositivo AWS Panorama. Para cada marco, utilice el formato compatible con SageMaker AI Neo y empaquete el modelo en un `.tar.gz` archivo.

Para obtener más información, consulte Compilar e implementar modelos con Neo en la Guía para desarrolladores de Amazon SageMaker AI.

Empaquetar un modelo

Un paquete de modelos consta de un descriptor, una configuración de paquete y un archivo de modelos. Al igual que en un paquete de imágenes de aplicaciones, la configuración del paquete indica al servicio AWS Panorama dónde se almacenan el modelo y el descriptor en Amazon S3.

Example packages/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**
>
> Especifique únicamente la versión principal y secundaria de la versión del framework. Para obtener una lista de las versiones compatibles PyTorch MXNet, de Apache y de las TensorFlow versiones, consulte Marcos compatibles.

Para importar un modelo, utilice el comando `import-raw-model` CLI de la aplicación AWS Panorama. Si realiza algún cambio en el modelo o en su descriptor, debe volver a ejecutar este comando para actualizar los activos de la aplicación. Para obtener más información, consulte Cambiar el modelo de visión artificial.

Para ver el esquema JSON del archivo descriptor, consulte assetDescriptor.schema.json.

Modelos de formación

Cuando entrene un modelo, use imágenes del entorno de destino o de un entorno de prueba que se parezca mucho al entorno de destino. Tenga en cuenta los siguientes factores que pueden afectar al rendimiento del modelo:

- Iluminación: la cantidad de luz que refleja un sujeto determina la cantidad de detalles que el modelo debe analizar. Es posible que un modelo entrenado con imágenes de sujetos bien iluminados no funcione bien en un entorno con poca luz o retroiluminado.
- Resolución: el tamaño de entrada de un modelo suele fijarse en una resolución de entre 224 y 512 píxeles de ancho en una relación de aspecto cuadrada. Antes de pasar un fotograma de vídeo al modelo, puede reducirlo o recortarlo para que se ajuste al tamaño requerido.
- Distorsión de la imagen: la distancia focal y la forma de la lente de la cámara pueden provocar que las imágenes se distorsionen alejándose del centro del encuadre. La posición de la cámara también determina qué características del sujeto son visibles. Por ejemplo, una cámara de techo con una lente gran angular mostrará la parte superior del sujeto cuando esté en el centro del encuadre y una vista sesgada del costado del sujeto a medida que se aleja del centro.

Para solucionar estos problemas, puede preprocesar las imágenes antes de enviarlas al modelo y entrenar al modelo sobre una variedad más amplia de imágenes que reflejen las variaciones de los entornos del mundo real. Si un modelo necesita funcionar en situaciones de iluminación y con una variedad de cámaras, necesitará más datos para el entrenamiento. Además de recopilar más imágenes, puede obtener más datos de entrenamiento creando variaciones de las imágenes existentes que estén sesgadas o tengan una iluminación diferente.

Creación de una imagen de aplicación

El dispositivo de AWS Panorama ejecuta aplicaciones como sistemas de archivos contenedores exportados a partir de una imagen que usted cree. Debe especificar las dependencias y los recursos de la aplicación en un Dockerfile que utiliza la imagen base de la aplicación de AWS Panorama como punto de partida.

Para crear una imagen de aplicación, utilice Docker y la CLI de aplicaciones de AWS Panorama. El siguiente ejemplo de la aplicación de muestra de esta guía muestra estos casos de uso.

Example packages/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
```

Se utilizan las siguientes instrucciones de Dockerfile.

- `FROM`: carga la imagen base de la aplicación (`public.ecr.aws/panorama/panorama-application`).
- `WORKDIR`: establece el directorio de trabajo en la imagen. `/panorama` se utiliza para el código de la aplicación y los archivos relacionados. Esta configuración solo se conserva durante la compilación y no afecta al directorio de trabajo de su aplicación en tiempo de ejecución (`/`).
- `COPY`: copia los archivos de una ruta local a una ruta de la imagen. `COPY . .` copia los archivos del directorio actual (el directorio del paquete) al directorio de trabajo de la imagen. Por ejemplo, el código de la aplicación se copia de `packages/123456789012-SAMPLE_CODE-1.0/application.py` a `/panorama/application.py`.
- `RUN`: ejecuta comandos del intérprete de comandos en la imagen durante la compilación. Una sola operación `RUN` puede ejecutar varios comandos en secuencia si se utiliza `&&` entre comandos. Este ejemplo actualiza el administrador de paquetes `pip` y, a continuación, instala las bibliotecas enumeradas en `requirements.txt`.

Puede utilizar otras instrucciones, como `ADD` y `ARG`, que resulten útiles en el momento de la compilación. Las instrucciones que añaden información de tiempo de ejecución al contenedor, por ejemplo `ENV`, no funcionan con AWS Panorama. AWS Panorama no ejecuta ningún contenedor

desde la imagen. Solo utiliza la imagen para exportar un sistema de archivos, que se transfiere al dispositivo.

Especificación de dependencias

`requirements.txt` es un archivo de requisitos de Python que especifica las bibliotecas utilizadas por la aplicación. La aplicación de ejemplo utiliza Open CV y AWS SDK para Python (Boto3).

Example packages/123456789012-SAMPLE_CODE-1.0/requirements.txt

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

El comando `pip install` del Dockerfile instala estas bibliotecas en el directorio `dist-packages` de Python que se encuentra debajo de `/usr/local/lib`, para que el código de su aplicación pueda importarlas.

Almacenamiento local

AWS Panorama reserva el directorio `/opt/aws/panorama/storage` para el almacenamiento de aplicaciones. Su aplicación puede crear y modificar archivos en esta ruta. Los archivos creados en el directorio de almacenamiento se conservan tras los reinicios. Las demás ubicaciones de los archivos temporales se borran al arrancar.

Creación de activos de imagen

Cuando crea una imagen para el paquete de aplicaciones con la CLI de aplicaciones de AWS Panorama, la CLI ejecuta `docker build` en el directorio del paquete. Esto crea una imagen de la aplicación que contiene el código de la aplicación. Luego, la CLI crea un contenedor, exporta su sistema de archivos, lo comprime y lo almacena en la carpeta `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
```

El bloque JSON de la salida es una definición de activo que la CLI agrega a la configuración del paquete (`package.json`) y registra en el servicio AWS Panorama. La CLI también copia el archivo descriptor, que especifica la ruta al script de la aplicación (el punto de entrada de la aplicación).

Example packages/123456789012-SAMPLE_CODE-1.0/descriptor.json

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

En la carpeta de activos, el descriptor y la imagen de la aplicación reciben el nombre de su suma de verificación SHA-256. Este nombre se utiliza como identificador único del activo cuando se almacena en Amazon S3.

Llamar a los servicios de AWS desde el código de su aplicación

Puede utilizarla AWS SDK for Python (Boto) para llamar a los servicios de AWS desde el código de su aplicación. Por ejemplo, si su modelo detecta algo fuera de lo común, puede publicar métricas en Amazon CloudWatch, enviar una notificación con Amazon SNS, guardar una imagen en Amazon S3 o invocar una función Lambda para su posterior procesamiento. La mayoría de los servicios de AWS tienen una API pública que puede usar con el SDK de AWS.

De forma predeterminada, el dispositivo no tiene permiso para acceder a ningún servicio de AWS. Para concederle permiso, cree un rol para la aplicación y asígnelo a la instancia de la aplicación durante la implementación.

Secciones

- Uso de Amazon S3
- Uso del tema MQTT AWS IoT

Uso de Amazon S3

Puede utilizar Amazon S3 para almacenar los resultados del procesamiento y otros datos de aplicación.

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

Uso del tema MQTT AWS IoT

Puede utilizar el SDK para Python (Boto3) para enviar mensajes a un tema de MQTT en AWS IoT. En el siguiente ejemplo, la aplicación publica en un tema que lleva el nombre del objeto del dispositivo y que se encuentra en la consola de 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")
```

Elija un nombre que indique el ID del dispositivo u otro identificador de su elección. Para publicar mensajes, la aplicación necesita permiso para llamar a `iot:Publish`.

Para supervisar una cola de MQTT

1. Abra la página Pruebas de la consola de AWS IoT.
2. Para Tema de suscripción, escriba un nombre para el tema. Por ejemplo, `panorama/panorama_my-appliance_Thing_a01e373b`.
3. Elija Suscribirse al tema.

El SDK de aplicaciones de AWS Panorama

El SDK de aplicaciones de AWS Panorama es una biblioteca de Python para desarrollar aplicaciones de AWS Panorama. En el código de su aplicación, usted utiliza el SDK de aplicaciones de AWS Panorama para cargar un modelo de visión artificial, ejecutar inferencias y enviar vídeo a un monitor.

> **Note**
>
> Para asegurarse de tener acceso a las funciones más recientes del SDK de aplicaciones de AWS Panorama, actualice el software del dispositivo.

Para obtener más información sobre las clases que define el SDK de la aplicación y sus métodos, consulte la referencia del SDK de la aplicación.

Secciones

- Añadir texto y cuadros a la salida de vídeo

Añadir texto y cuadros a la salida de vídeo

Con el SDK de AWS Panorama, puede enviar una transmisión de vídeo a una pantalla. El vídeo puede incluir texto y cuadros que muestren el resultado del modelo, el estado actual de la aplicación u otros datos.

Cada objeto de la matriz `video_in` es una imagen de una transmisión de cámara que está conectada al dispositivo. El tipo de este objeto es `panoramasdk.media`. Tiene métodos para añadir texto y cuadros rectangulares a la imagen, que luego puede asignar a la matriz `video_out`.

En el ejemplo siguiente, la aplicación de muestra añade una etiqueta para cada uno de los resultados. Cada resultado se coloca en la misma posición a la izquierda, pero a diferentes alturas.

```
        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)
```

Para añadir un cuadro a la imagen de salida, utilice `add_rect`. Este método toma 4 valores entre 0 y 1, indicando la posición de las esquinas superior izquierda e inferior derecha del cuadro.

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

Ejecutar múltiples subprocesos

Puede ejecutar la lógica de la aplicación en un subproceso de procesamiento y utilizar otros subprocesos para otros procesos en segundo plano. Por ejemplo, puede crear un hilo que sirva al tráfico HTTP para la depuración o un hilo que supervise los resultados de las inferencias y envíe datos a. AWS

Para ejecutar varios subprocesos, utilice el módulo de subprocesos de la biblioteca estándar de Python para crear un subproceso para cada proceso. El siguiente ejemplo muestra el bucle principal de la aplicación de muestra del servidor de depuración, que crea un objeto de aplicación y lo utiliza para ejecutar tres subprocesos.

Example packages/123456789012-DEBUG_SERVER-1.0/application.py: bucle principal

```
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.')
```

Cuando se cierran todos los subprocesos, la aplicación se reinicia automáticamente. El bucle `run_cv` procesa las imágenes de las transmisiones de la cámara. Si recibe una señal para

detenerse, cierra el proceso de depuración, lo cual ejecuta un servidor HTTP y no puede apagarse solo. Cada subproceso debe gestionar sus propios errores. Si no se detecta ni registra un error, el subproceso se cierra silenciosamente.

Example packages/123456789012-DEBUG_SERVER-1.0/application.py: bucle de procesamiento

```
    # 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")
```

Los subprocesos se comunican a través del objeto `self` de la aplicación. Para reiniciar el ciclo de procesamiento de la aplicación, el subproceso del depurador llama al método `stop`. Este método establece un atributo de `terminate` que indica a los demás subprocesos que se cierren.

Example packages/123456789012-DEBUG_SERVER-1.0/application.py: método de parada

```
    # 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)
```

Ofrecer servicios al tráfico de entrada

Puede monitorizar o depurar aplicaciones de forma local ejecutando un servidor HTTP junto con el código de la aplicación. Para atender el tráfico externo, asigne los puertos del dispositivo de AWS Panorama a los puertos del contenedor de aplicaciones.

> **Important**
>
> De forma predeterminada, el dispositivo de AWS Panorama no acepta tráfico entrante en ningún puerto. Abrir puertos en el dispositivo conlleva un riesgo de seguridad implícito. Al utilizar esta característica, debe tomar medidas adicionales para proteger el dispositivo del tráfico externo y proteger las comunicaciones entre los clientes autorizados y el dispositivo. El código de muestra incluido en esta guía tiene fines de demostración y no implementa la autenticación, la autorización ni el cifrado.

Puede abrir puertos en el rango de 8000 a 9000 en el dispositivo. Estos puertos, una vez abiertos, pueden recibir tráfico de cualquier cliente enrutable. Al implementar la aplicación, debe especificar qué puertos abrir, y asignar los puertos del dispositivo a los puertos del contenedor de aplicaciones. El software del dispositivo reenvía el tráfico al contenedor y envía las respuestas al solicitante. Las solicitudes se reciben en el puerto del dispositivo que especifique y las respuestas se envían a un puerto efímero al azar.

Configuración de puertos de entrada

Las asignaciones de puertos se especifican en tres lugares de la configuración de la aplicación. En el paquete de códigos `package.json`, usted especifica el puerto que escucha el nodo de código en un bloque de `network`. El siguiente ejemplo declara que el nodo escucha en el puerto 80.

Example packages/123456789012-DEBUG_SERVER-1.0/package.json

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

```
                    {
                        "port": 80,
                        "description": "http"
                    }
                ]
            }
```

En el manifiesto de la aplicación, se declara una regla de enrutamiento que asigna un puerto del dispositivo a un puerto del contenedor de códigos de la aplicación. El siguiente ejemplo agrega una regla que asigna el puerto 8080 del dispositivo al puerto 80 del contenedor de `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."
            }
        }
    ]
```

Al implementar la aplicación, se especifican las mismas reglas en la consola de AWS Panorama o se transfiere un documento de anulación a la CreateApplicationInstanceAPI. Debe proporcionar esta configuración en el momento de la implementación para confirmar que desea abrir los puertos del dispositivo.

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"
    }
}
```

Si otra aplicación utiliza el puerto del dispositivo especificado en el manifiesto de la aplicación, puedes usar el documento de anulación para elegir otro puerto.

Ofrecer servicios al tráfico

Con los puertos abiertos en el contenedor, puede abrir un socket o ejecutar un servidor para gestionar las solicitudes entrantes. El ejemplo de `debug-server` muestra una implementación básica de un servidor HTTP que se ejecuta junto con el código de una aplicación de visión artificial.

> **Important**
>
> La implementación de ejemplo no es segura para su uso en producción. Para evitar que su dispositivo sea vulnerable a los ataques, debe implementar los controles de seguridad adecuados en la configuración de código y red.

Example packages/123456789012-DEBUG_SERVER-1.0/Application.py: servidor 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.')
```

El servidor acepta solicitudes GET en la ruta de `/status` para recuperar cierta información sobre la aplicación. También acepta una solicitud POST a `/restart` para reiniciar la aplicación.

Para demostrar esta funcionalidad, la aplicación de ejemplo ejecuta un cliente HTTP en un hilo independiente. El cliente llama a la ruta de `/status` a través de la red local poco después del inicio y reinicia la aplicación unos minutos más tarde.

Example packages/123456789012-DEBUG_SERVER-1.0/Application.py: cliente 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")
```

El circuito principal gestiona los subprocesos y reinicia la aplicación cuando se cierran.

Example packages/123456789012-DEBUG_SERVER-1.0/Application.py: bucle principal

```
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.')
```

Para implementar la aplicación de muestra, consulte las instrucciones del GitHub repositorio de esta guía.

Uso de la GPU

Puede acceder al procesador de gráficos (GPU) del dispositivo de AWS Panorama para usar bibliotecas aceleradas por GPU o ejecutar modelos de machine learning en el código de la aplicación. Para activar el acceso a la GPU, añada el acceso a la GPU como requisito a la configuración del paquete después de crear el contenedor de código de la aplicación.

> **Important**
>
> Si habilita el acceso a la GPU, no podrá ejecutar nodos modelo en ninguna aplicación en el dispositivo. Por motivos de seguridad, el acceso a la GPU está restringido cuando el dispositivo ejecuta un modelo compilado con SageMaker AI Neo. Con el acceso a la GPU, debe ejecutar sus modelos en nodos de código de aplicación y todas las aplicaciones en el dispositivo comparten el acceso a la GPU.

Para activar el acceso a la GPU de su aplicación, actualice la configuración del paquete después de crearlo con la CLI de aplicaciones de AWS Panorama. En el siguiente ejemplo, se muestra el bloque `requirements` que añade el acceso de la GPU al nodo de código de la aplicación.

Example package.json con bloque de requisitos

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

Actualice la configuración del paquete entre los pasos de creación y empaquetado de su flujo de trabajo.

Para implementar una aplicación con acceso a la GPU

1. Para crear el contenedor de aplicaciones, utilice el comando `build-container`.

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

2. Añada el bloque `requirements` a la configuración del paquete.
3. Para cargar la configuración del paquete y el activo del contenedor, utilice el comando `package-application`.

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

4. Implemente la aplicación .

Para ver ejemplos de aplicaciones que utilizan el acceso a la GPU, visite el aws-panorama-samples GitHub repositorio.

Configuración de un entorno de desarrollo en Windows

Para crear una aplicación de AWS Panorama, debe utilizar Docker, herramientas de línea de comandos y Python. En Windows, puede configurar un entorno de desarrollo mediante Docker Desktop con el subsistema de Windows para Linux y Ubuntu. En este tutorial, se explica el proceso de configuración de un entorno de desarrollo que se ha probado con las herramientas y aplicaciones de muestra de AWS Panorama.

Secciones

Requisitos previos

Para seguir este tutorial, necesita una versión de Windows que sea compatible con el subsistema de Windows para Linux 2 (WSL 2).

- Windows 10 versión 1903 y superior (compilación 18362 y superior) o Windows 11
- Características de Windows
  - Windows Subsystem for Linux
  - Hyper-V
  - Plataforma de máquinas virtuales

Este tutorial se desarrolló con las siguientes versiones de software.

- Ubuntu 20.04
- Python 3.8.5
- Docker 20.10.8

Instalar WSL 2 y Ubuntu

Si tiene Windows 10 versión 2004 o superior (compilación 19041 y superior), puede instalar WSL 2 y Ubuntu 20.04 con el siguiente comando. PowerShell

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

Para versiones anteriores de Windows, siga las instrucciones de la documentación de WSL 2: Pasos de la instalación manual para versiones anteriores

Instalar Docker

Para instalar Docker Desktop, descargue y ejecute el paquete de instalación desde hub.docker.com. Si tiene problemas, siga las instrucciones del sitio web de Docker: Docker Desktop WSL 2 backend.

Ejecute Docker Desktop y siga el tutorial de primera ejecución para crear un contenedor de ejemplo.

> **Note**
>
> Docker Desktop solo habilita Docker en la distribución predeterminada. Si tiene otras distribuciones de Linux instaladas antes de ejecutar este tutorial, habilite Docker en la distribución de Ubuntu recién instalada en el menú de configuración de Docker Desktop, en Recursos, integración con WSL.

Configurar Ubuntu

Ahora puede ejecutar los comandos de Docker en su máquina virtual Ubuntu. Para abrir un terminal de línea de comandos, ejecute la distribución desde el menú de inicio. La primera vez que la ejecute, configurará un nombre de usuario y una contraseña que podrá utilizar para ejecutar comandos de administrador.

Para completar la configuración de su entorno de desarrollo, actualice el software de la máquina virtual e instale las herramientas.

Para configurar la máquina virtual

1. Actualice el software que viene con Ubuntu.

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

2. Instale las herramientas de desarrollo con apt.

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

3. Instale las bibliotecas de Python con pip.

```
$ pip3 install awscli panoramacli
```

4. Abra una nueva terminal y, a continuación, ejecute `aws configure` para configurar AWS CLI.

```
$ aws configure
```

   Si no tiene claves de acceso, puede generarlas en la consola de IAM.

Por último, descargue e importe la aplicación de muestra.

Para obtener la aplicación de muestra

1. Descargue y extraiga la aplicación de muestra.

```
$ 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. Ejecute los scripts incluidos para probar la compilación, crear el contenedor de aplicaciones y cargar los paquetes en AWS Panorama.

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

La CLI de aplicaciones de AWS Panorama carga los paquetes y los registra en el servicio AWS Panorama. Ahora puede implementar la aplicación de muestra con la consola de AWS Panorama.

Pasos a seguir a continuación

Para explorar y editar los archivos del proyecto, puede utilizar el explorador de archivos o un entorno de desarrollo integrado (IDE) compatible con WSL.

Para acceder al sistema de archivos de la máquina virtual, abra el explorador de archivos y escriba `\\wsl$` en la barra de navegación. Este directorio contiene un enlace al sistema de archivos de la máquina virtual (`Ubuntu-20.04`) y a los sistemas de archivos de los datos de Docker. En `Ubuntu-20.04`, su directorio de usuarios está en `home\`*`username`*.

> **Note**
>
> Para acceder a los archivos de su instalación de Windows desde Ubuntu, navegue hasta el directorio `/mnt/c`. Por ejemplo, puede ver una lista de los archivos de su directorio de descargas ejecutando `ls /mnt/c/Users/`*`windows-username`*`/Downloads`.

Con Visual Studio Code, puede editar el código de la aplicación en su entorno de desarrollo y ejecutar comandos con una terminal integrada. Para instalar Visual Studio Code, visite code.visualstudio.com. Tras la instalación, añada la extensión Remote WSL.

La terminal de Windows es una alternativa a la terminal estándar de Ubuntu en la que ha estado ejecutando comandos. Admite múltiples pestañas y puede ejecutar PowerShell, línea de comandos y terminales para cualquier otra variedad de Linux que instale. Es compatible con las funciones de copiar y pegar con Ctrl+C los Ctrl+V botones «y hacer clic» URLs, entre otras mejoras útiles. Para instalar la terminal de Windows, visite microsoft.com.

La API de AWS Panorama

Puede usar la API pública del servicio AWS Panorama para automatizar los flujos de trabajo de administración de dispositivos y aplicaciones. Con el AWS Command Line Interface o el AWS SDK, puede desarrollar scripts o aplicaciones que gestionen los recursos y las implementaciones. El GitHub repositorio de esta guía incluye scripts que puede usar como punto de partida para su propio código.

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

Secciones

Automatizar el registro de dispositivos

Para aprovisionar un dispositivo, utilice la ProvisionDeviceAPI. La respuesta incluye un archivo ZIP con la configuración del dispositivo y las credenciales temporales. Decodifique el archivo y guárdelo en un archivo con el prefijo `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}"
```

Las credenciales del archivo de configuración caducan a los 5 minutos. Transfiera el archivo a su dispositivo con la unidad USB incluida.

Para registrar una cámara, utilice la CreateNodeFromTemplateJobAPI. Esta API toma un mapa de los parámetros de la plantilla para el nombre de usuario, la contraseña y la URL de la cámara. Puede formatear este mapa como un documento JSON mediante la manipulación de cadenas 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)
```

Como alternativa, puede cargar la configuración JSON desde un archivo.

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

Administrar dispositivos con la API de AWS Panorama

Puede automatizar las tareas de administración de dispositivos con la API de AWS Panorama.

Ver dispositivos

Para obtener una lista de dispositivos con dispositivo IDs, utilice la ListDevicesAPI.

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

Para obtener más información sobre un dispositivo, utilice la 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
}
```

Actualizar el software de dispositivo

Si la versión `LatestSoftware` es más reciente que la `CurrentSoftware`, puede actualizar el dispositivo. Utilice la CreateJobForDevicesAPI para crear un trabajo de actualización 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"
        }
    ]
}
```

En un script, puede rellenar el campo de versión de la imagen en el archivo de configuración del trabajo manipulando cadenas 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
}
```

El dispositivo descarga la versión de software especificada y se actualiza automáticamente. Observe el progreso de la actualización con la 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
}
```

Para obtener una lista de todos los trabajos en ejecución, usa la ListDevicesJobs.

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

Para ver un ejemplo de script que busca y aplica las actualizaciones, consulte el archivo check-updates.sh en el GitHub repositorio de esta guía.

Reinicio de dispositivos

Para reiniciar un dispositivo, utilice la CreateJobForDevicesAPI.

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

```
    ]
}
```

En un script, puede obtener una lista de dispositivos y elegir uno para reanudarlo de forma interactiva.

Example reboot-device.sh: uso

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

Automatización de las implementaciones de aplicaciones

Para implementar una aplicación, debe utilizar la CLI de aplicaciones AWS Panorama y AWS Command Line Interface. Tras crear el contenedor de la aplicación, debe cargar este y otros recursos a un punto de acceso de Amazon S3. A continuación, debe implementar la aplicación con la CreateApplicationInstanceAPI.

Para obtener más contexto e instrucciones sobre el uso de los scripts que se muestran, siga las instrucciones del README de la aplicación de ejemplo.

Secciones

- Cree el contenedor
- Cargue el contenedor y registre los nodos
- Implemente de la aplicación
- Monitorice la implementación

Cree el contenedor

Para crear el contenedor de aplicaciones, utilice el comando `build-container`. Este comando crea un contenedor de Docker y lo guarda como un sistema de archivos comprimido en la carpeta `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
```

También puede utilizar la opción de completar la línea de comandos para rellenar el argumento de la ruta; para ello, escriba parte de la ruta y, a continuación, presione TAB.

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

Cargue el contenedor y registre los nodos

Para cargar la aplicación, utilice el comando `package-application`. Este comando carga recursos de la carpeta de `assets` a un punto de acceso de Amazon S3 que administra AWS Panorama.

Example 4-package-app.sh

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

La CLI de la aplicación AWS Panorama carga los activos de contenedores y descriptores a los que hace referencia la configuración del paquete (`package.json`) en cada paquete y los registra como nodos en AWS Panorama. A continuación, consulte estos nodos en el manifiesto de la aplicación (`graph.json`) para implementar la aplicación.

Implemente de la aplicación

Para implementar la aplicación, se usa la CreateApplicationInstanceAPI. Esta acción requiere, entre otros, los siguientes parámetros.

- `ManifestPayload`: el manifiesto de la aplicación (`graph.json`) que define los nodos, paquetes, periferias y parámetros de la aplicación.
- `ManifestOverridesPayload`: un segundo manifiesto que anula los parámetros del primero. El manifiesto de la aplicación se puede considerar un recurso estático en la fuente de la aplicación, mientras que el manifiesto de anulación proporciona ajustes en el momento de la implementación que personalizan la implementación.
- `DefaultRuntimeContextDevice`: el dispositivo de destino.
- `RuntimeRoleArn`: el ARN de un rol de IAM que la aplicación utiliza para acceder a los servicios y recursos de AWS.
- `ApplicationInstanceIdToReplace`: el ID de una instancia de aplicación existente que se va a eliminar del dispositivo.

Las cargas útiles de manifiesto y anulación son documentos JSON que deben proporcionarse como un valor de cadena anidado dentro de otro documento. Para ello, el script carga los manifiestos de un archivo en forma de cadena y utiliza la herramienta jq para construir el documento anidado.

Example 5-deploy.sh: redacta manifiestos

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

El script de despliegue utiliza la ListDevicesAPI para obtener una lista de los dispositivos registrados en la región actual y guarda la elección del usuario en un archivo local para despliegues posteriores.

Example 5-deploy.sh: busca 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)
```

El rol de la aplicación se crea mediante otro script (1-create-role.sh). El script de despliegue obtiene el ARN de este rol. AWS CloudFormation Si la aplicación ya está implementada en el dispositivo, el script obtiene el ID de esa instancia de la aplicación de un archivo local.

Example 5-deploy.sh: ARN del rol y argumentos de reemplazo

```
# 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
```

Por último, el script reúne todas las piezas para crear una instancia de aplicación e implementar la aplicación en el dispositivo. El servicio responde con un ID de instancia que el script almacena para su uso posterior.

Example 5-deploy.sh: implementa la aplicación

```
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
```

Monitorice la implementación

Para supervisar una implementación, usa la ListApplicationInstancesAPI. El script del monitor obtiene el ID del dispositivo y el ID de la instancia de la aplicación de los archivos del directorio de la aplicación y los utiliza para construir un comando CLI. A continuación, realiza una llamada en bucle.

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
```

Cuando se complete la implementación, podrá ver los registros llamando a la API de Amazon CloudWatch Logs. El script de visualización de registros utiliza la `GetLogEvents` API CloudWatch Logs.

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
```

Administre las aplicaciones con la API de AWS Panorama

Puede monitorear y administrar aplicaciones con la API de AWS Panorama.

Ver aplicaciones

Para obtener una lista de las aplicaciones que se ejecutan en un dispositivo, utilice la 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"
            }
        },
    ]
}
```

Para obtener más información sobre los nodos de una instancia de aplicación, utilice la 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"
        }
    ]
}
```

Gestione las transmisiones de cámara

Puedes pausar y reanudar los nodos de transmisión de cámara con la SignalApplicationInstanceNodeInstancesAPI.

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

En un script, puede obtener una lista de nodos y elegir uno para pausarlo o reanudarlo de forma interactiva.

Example pause-camera.sh — uso

```
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"
}
```

Al pausar y reanudar los nodos de la cámara, puede recorrer un número mayor de transmisiones de cámara de las que se pueden procesar simultáneamente. Para ello, asigne varias secuencias de cámara al mismo nodo de entrada de su manifiesto de anulación.

En el siguiente ejemplo, el manifiesto de anulación asigna dos secuencias de cámara `warehouse-floor` y `entrance-north` al mismo nodo de entrada (`camera_node`). La transmisión `warehouse-floor` está activa cuando se inicia la aplicación y el nodo `entrance-north` espera a que se encienda una señal.

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"
                    }
                ]
            }
```

Para obtener más información sobre la implementación con la API, consulte Automatización de las implementaciones de aplicaciones.

Uso de puntos de conexión de VPC

Si trabaja en una VPC sin acceso a Internet, puede crear un punto de conexión de VPC para usarlo con AWS Panorama. Un punto de conexión de VPC permite a los clientes que se ejecutan en una subred privada conectarse a un servicio de AWS sin conexión a Internet.

Para obtener información detallada sobre los puertos y puntos de conexión que utiliza el dispositivo de AWS Panorama, consulte ???.

Secciones

- Creación de un punto de conexión de VPC
- Conexión de un dispositivo a una subred privada
- Plantillas de muestra AWS CloudFormation

Creación de un punto de conexión de VPC

Para establecer una conexión privada entre su VPC y AWS Panorama, cree un punto de conexión de VPC. No se requiere un punto de conexión de VPC para usar AWS Panorama. Solo necesita crear un punto de conexión de VPC si trabaja en una VPC sin acceso a Internet. Cuando la CLI o el SDK de AWS intentan conectarse a AWS Panorama, el tráfico se enruta a través del punto de conexión de VPC.

Cree un punto de conexión de VPC para AWS Panorama con la siguiente configuración:

- Nombre de servicio: **`com.amazonaws.`*`us-west-2`*`.panorama`**
- Tipo: interfaz

Un punto de conexión de VPC usa el nombre DNS del servicio para obtener tráfico de los clientes del SDK de AWS sin necesidad de realizar ninguna configuración adicional. Para obtener más información sobre el uso de los puntos de conexión de VPC, consulte Puntos de enlace de la VPC de tipo interfaz en la Guía del usuario de Amazon VPC.

Conexión de un dispositivo a una subred privada

El dispositivo AWS Panorama se puede conectar a AWS través de una conexión VPN privada con AWS Site-to-Site VPN o AWS Direct Connect. Con estos servicios, puede crear una subred privada

que se extienda hasta su centro de datos. El dispositivo se conecta a la subred privada y accede a los AWS servicios a través de los puntos de enlace de la VPC.

Site-to-Site VPN y AWS Direct Connect son servicios para conectar su centro de datos a Amazon VPC de forma segura. Con la Site-to-Site VPN, puede utilizar dispositivos de red disponibles en el mercado para conectarse. AWS Direct Connect utiliza un AWS dispositivo para conectarse.

- Site-to-Site VPN: ¿qué es AWS Site-to-Site VPN?
- AWS Direct Connect – ¿Qué es AWS Direct Connect?

Después de conectar la red local a una subred privada en una VPC, cree puntos de conexión de VPC para los siguientes servicios.

- Amazon Simple Storage Service: AWS PrivateLink para Amazon S3
- AWS IoT Core: uso de AWS IoT Core con puntos de conexión de VPC de interfaz (plano de datos y proveedor de credenciales)
- Amazon Elastic Container Registry: puntos de conexión de VPC de la interfaz de Amazon Elastic Container Registry
- Amazon CloudWatch: uso CloudWatch con puntos de enlace de VPC de interfaz
- Amazon CloudWatch Logs: uso de CloudWatch registros con puntos de enlace de VPC de interfaz

El dispositivo no necesita conectividad con el servicio AWS Panorama. Se comunica con AWS Panorama a través de un canal de mensajería en AWS IoT.

Además de los puntos de enlace de VPC, Amazon S3 y Amazon AWS IoT requieren el uso de zonas alojadas privadas de Amazon Route 53. La zona alojada privada dirige el tráfico de los subdominios, incluidos los subdominios de los puntos de acceso de Amazon S3 y los temas de MQTT, al punto de conexión de VPC correcto. Para obtener información sobre las zonas alojadas privadas, consulte Trabajar con zonas alojadas privadas en la Guía para desarrolladores de Amazon Route 53.

Para ver un ejemplo de configuración de VPC con puntos de conexión de VPC y zonas alojadas privadas, consulte Plantillas de muestra AWS CloudFormation.

Plantillas de muestra AWS CloudFormation

El GitHub repositorio de esta guía proporciona AWS CloudFormation plantillas que puede usar para crear recursos para usarlos con AWS Panorama. Las plantillas crean una VPC con dos subredes privadas, una subred pública y un punto de conexión de VPC. Puede usar las subredes privadas de la VPC para alojar recursos aislados de Internet. Los recursos de la subred pública pueden comunicarse con los recursos privados, pero no se puede acceder a los recursos privados desde Internet.

Example vpc-endpoint.yml: subredes privadas

```
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
  ...
```

En la plantilla de `vpc-endpoint.yml`, se muestra cómo crear un punto de conexión de VPC para AWS Panorama. Puede usar este punto de conexión para administrar los recursos de AWS Panorama con el AWS SDK o AWS CLI.

Example vpc-endpoint.yml: punto de conexión de 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` es una política de permisos basada en recursos que define las llamadas a la API que se pueden realizar con el punto de conexión. Puede modificar la política para restringir las acciones y los recursos a los que se puede acceder a través del punto de conexión. Para más información, consulte Control del acceso a los servicios con puntos de conexión de VPC en la Guía del usuario de Amazon VPC.

La plantilla de `vpc-appliance.yml` muestra cómo crear puntos de conexión de VPC y zonas alojadas privadas para los servicios que utiliza el dispositivo de AWS Panorama.

Example vpc-appliance.yml: punto de conexión de los puntos de acceso de Amazon S3 con zona alojada privada

```
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 ] ] ]
```

En las plantillas de ejemplo, se muestra la creación de los recursos de Amazon VPC y Route 53 con una VPC de muestra. Puede adaptarlos a su caso de uso eliminando los recursos de la VPC y sustituyendo las referencias a la subred, el grupo de seguridad y la VPC IDs por las de sus recursos. IDs

Ejemplos de aplicaciones, scripts y plantillas

El GitHub repositorio de esta guía proporciona ejemplos de aplicaciones, scripts y plantillas para AWS Panorama dispositivos. Utilice estos ejemplos para aprender las prácticas recomendadas y automatizar los flujos de trabajo de desarrollo.

Secciones

Aplicaciones de muestra

Los ejemplos de aplicaciones muestran el uso de AWS Panorama funciones y tareas comunes de visión artificial. Estas aplicaciones de muestra incluyen scripts y plantillas que automatizan la configuración y la implementación. Con una configuración mínima, puede implementar y actualizar aplicaciones desde la línea de comandos.

- aws-panorama-sample— Visión artificial básica con un modelo de clasificación. Úselo AWS SDK for Python (Boto) para cargar métricas CloudWatch, instrumentar métodos de preprocesamiento e inferencia y configurar el registro.
- servidor de depuración: abra los puertos entrantes del dispositivo y reenvíe el tráfico a un contenedor de códigos de aplicación. Utilice subprocesos múltiples para ejecutar código de aplicación, un servidor HTTP y un cliente HTTP de forma simultánea.
- modelo personalizado: exporte modelos desde el código y compílelos con SageMaker AI Neo para comprobar la compatibilidad con el dispositivo. AWS Panorama Compile localmente en un desarrollo de Python, en un contenedor de Docker o en una EC2 instancia de Amazon. Exporte y compile todos los modelos de aplicaciones integrados en Keras para una versión específica TensorFlow o de Python.

Para ver más ejemplos de aplicaciones, visite también el aws-panorama-samplesrepositorio.

Scripts de utilidades

Los scripts del `util-scripts` directorio administran los AWS Panorama recursos o automatizan los flujos de trabajo de desarrollo.

- provision-device.sh: aprovisione un dispositivo.
- check-updates.sh: compruebe si hay actualizaciones de software del dispositivo y aplíquelas.
- reboot-device.sh: reinicie un dispositivo.
- register-camera.sh: registre una cámara.
- deregister-camera.sh: elimine un nodo de cámara.
- view-logs.sh: vea los registros de una instancia de aplicación.
- pause-camera.sh: pause o reanude la transmisión de una cámara.
- push.sh: cree, cargue e implemente una aplicación.
- rename-package.sh: cambie el nombre de un paquete de nodos. Actualiza los nombres de los directorios, los archivos de configuración y el manifiesto de la aplicación.
- samplify.sh: sustituya su ID de cuenta por un ID de cuenta de ejemplo y restaure las configuraciones de respaldo para eliminar la configuración local.
- update-model-config.sh — Vuelva a añadir el modelo a la aplicación después de actualizar el archivo descriptor.
- cleanup-patches.sh: anule el registro de las versiones de parches antiguas y elimine sus manifiestos de Amazon S3.

Para obtener información sobre el uso, consulte el archivo README.

AWS CloudFormation plantillas

Utilice las AWS CloudFormation plantillas del `cloudformation-templates` directorio para crear recursos para AWS Panorama las aplicaciones.

- alarm-application.yml: cree una alarma que supervise una aplicación en busca de errores. Si la instancia de la aplicación genera errores o deja de ejecutarse durante 5 minutos, la alarma envía una notificación por correo electrónico.

- alarm-device.yml: cree una alarma que supervise la conectividad de un dispositivo. Si el dispositivo deja de enviar métricas durante 5 minutos, la alarma envía una notificación por correo electrónico.
- application-role.yml: cree un rol de aplicación. El rol incluye permiso para enviar métricas a CloudWatch. Añada permisos a la declaración de política para otras operaciones de API que utilice su aplicación.
- vpc-appliance.yml: cree una VPC con acceso al servicio de subred privado para el dispositivo. AWS Panorama Para conectar el dispositivo a una VPC, utilice AWS Direct Connect o. AWS Site-to-Site VPN
- vpc-endpoint.yml: cree una VPC con acceso al servicio de subred privada. AWS Panorama Los recursos de la VPC se pueden conectar AWS Panorama para supervisar y gestionar AWS Panorama los recursos sin necesidad de conectarse a Internet.

El `create-stack.sh` script de este directorio crea AWS CloudFormation pilas. Requiere un número variable de argumentos. El primer argumento es el nombre de la plantilla y los argumentos restantes sustituyen a los parámetros de la plantilla.

Por ejemplo, el siguiente comando crea un rol de aplicación.

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

Más ejemplos y herramientas

El aws-panorama-samplesrepositorio tiene más aplicaciones de muestra y herramientas útiles.

- Aplicaciones: aplicaciones de muestra para diversas arquitecturas de modelos y casos de uso.
- Validación del flujo de cámara: valide los flujos de cámara.
- PanoJupyter— Se ejecuta JupyterLab en un AWS Panorama dispositivo.
- Transferencia local: actualice el código de la aplicación sin crear ni implementar un contenedor de aplicaciones.

La AWS comunidad también ha desarrollado herramientas y directrices para AWS Panorama. Consulte los siguientes proyectos de código abierto en GitHub.

- cookiecutter-panorama: una plantilla de Cookiecutter para aplicaciones. AWS Panorama

- backpack: módulos de Python para acceder a los detalles del entorno del tiempo de ejecución, la creación de perfiles y las opciones adicionales de salida de vídeo.

Supervisión de AWS Panorama recursos y aplicaciones

Puedes monitorizar AWS Panorama los recursos en la AWS Panorama consola y con Amazon CloudWatch. El AWS Panorama dispositivo se conecta a la AWS nube a través de Internet para informar sobre su estado y el estado de las cámaras conectadas. Mientras está encendido, el dispositivo también envía CloudWatch registros a Logs en tiempo real.

El dispositivo obtiene permiso para usar AWS IoT CloudWatch los registros y otros servicios de AWS a partir de un rol de servicio que usted crea la primera vez que usa la AWS Panorama consola. Para obtener más información, consulte Roles de servicio y recursos multiservicios de AWS Panorama.

Si necesita ayuda para solucionar errores específicos, consulte Solución de problemas.

Temas

Supervisión en la consola de AWS Panorama

Puede usar la consola de AWS Panorama para supervisar su dispositivo y sus cámaras de AWS Panorama. La consola se utiliza AWS IoT para monitorear el estado del dispositivo.

Para supervisar su dispositivo en la consola de AWS Panorama

1. Abra la consola de AWS Panorama.
2. Abra la página Dispositivos de la consola de AWS Panorama.
3. Elija un dispositivo.
4. Para ver el estado de una instancia de aplicación, selecciónela en la lista.
5. Para ver el estado de las interfaces de red del dispositivo, seleccione Configuración.

El estado general del dispositivo aparece en la parte superior de la página. Si el estado es En línea, el dispositivo está conectado a las actualizaciones de estado periódicas AWS y las envía.

Visualización de los registros de AWS Panorama

AWS Panorama informa de los eventos de las aplicaciones y del sistema a Amazon CloudWatch Logs. Cuando tenga problemas, puede utilizar los registros de eventos para depurar su aplicación de AWS Panorama o solucionar problemas de configuración de la aplicación.

Para ver los registros en CloudWatch Logs

1. Abra la página de grupos de registros de la consola de CloudWatch registros.
2. Encuentre los registros de aplicaciones y dispositivos de AWS Panorama en los siguientes grupos:

   - Registros de dispositivos: `/aws/panorama/devices/`*`device-id`*
   - Registros de aplicaciones: `/aws/panorama/devices/`*`device-id`*`/applications/`*`instance-id`*

Al volver a aprovisionar un dispositivo después de actualizar el software del sistema, también puede ver los registros en la unidad USB de aprovisionamiento.

Secciones

Visualización de los registros del dispositivo

El dispositivo de AWS Panorama crea un grupo de registros para el dispositivo y un grupo para cada instancia de aplicación que implemente. Los registros del dispositivo contienen información sobre el estado de la aplicación, las actualizaciones del software y la configuración del sistema.

Registros de dispositivos: **`/aws/panorama/devices/`*`device-id`***

- `occ_log`: salida del proceso del controlador. Este proceso coordina las implementaciones de las aplicaciones e informa sobre el estado de los nodos de cada instancia de la aplicación.

- `ota_log`— Resultado del proceso que coordina las actualizaciones de software over-the-air (OTA).
- `syslog`: salida del proceso syslog del dispositivo, que captura los mensajes enviados entre procesos.
- `kern_log`: eventos del núcleo de Linux del dispositivo.
- `logging_setup_logs`— Resultado del proceso que configura el agente CloudWatch Logs.
- `cloudwatch_agent_logs`— Salida del agente de CloudWatch registros.
- `shadow_log`: salida de la AWS IoT sombra de dispositivo.

Visualización de registros de aplicaciones

El grupo de registros de una instancia de aplicación contiene un flujo de registro para cada nodo, que lleva el nombre del nodo.

Registros de aplicaciones: **`/aws/panorama/devices/`*`device-id`*`/applications/`*`instance-id`***

- Código: salida del código de su aplicación y del SDK de aplicaciones de AWS Panorama. Agrega los registros de las aplicaciones de `/opt/aws/panorama/logs`.
- Modelo: salida del proceso que coordina las solicitudes de inferencia con un modelo.
- Transmisión: salida del proceso que decodifica el vídeo de una secuencia de cámara.
- Pantalla: salida del proceso que renderiza la salida de vídeo para el puerto HDMI.
- `mds`: registros del servidor de metadatos del dispositivo.
- `console_output`: captura flujos de error y salida estándar de los contenedores de código.

Si no ve los CloudWatch registros en Logs, confirme que se encuentra en la región de AWS correcta. Si es así, es posible que haya un problema con la conexión del dispositivo a AWS o con los permisos de la función del dispositivo AWS Identity and Access Management (IAM).

Configuración de registros de aplicaciones

Configure un registrador de Python para escribir archivos de registro en `/opt/aws/panorama/logs`. El dispositivo transmite los registros de esta ubicación a CloudWatch Logs. Para evitar ocupar demasiado espacio en disco, utilice un tamaño de archivo máximo de 10 MiB y un número de copias de seguridad de 1. En el siguiente ejemplo, se muestra un método que crea un registrador.

Example application.py: configuración del registrador

```
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
```

Inicialice el registrador en el ámbito global y utilícelo en todo el código de la aplicación.

Example application.py: inicializar el registrador

```
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()
```

Visualización de registros de aprovisionamiento

Durante el aprovisionamiento, el dispositivo de AWS Panorama copia los registros en la unidad USB que utilice para transferir el archivo de configuración al dispositivo. Utilice estos registros para solucionar problemas de aprovisionamiento en los dispositivos con la versión de software más reciente.

> **Important**
>
> Los registros de aprovisionamiento están disponibles para los dispositivos actualizados a la versión de software 4.3.23 o posterior.

Registros de aplicaciones

- `/panorama/occ.log`: registros del software del controlador de AWS Panorama.
- `/panorama/ota_agent.log`— Registros de agentes de over-the-air actualización de AWS Panorama.
- `/panorama/syslog.log`: registros del sistema Linux.
- `/panorama/kern.log`: registros de kernel de Linux.

Registros de salida de un dispositivo

Si los registros de su dispositivo y aplicación no aparecen en CloudWatch Logs, puede utilizar una unidad USB para extraer una imagen de registro cifrada del dispositivo. El equipo de servicio AWS Panorama puede descifrar los registros en su nombre y ayudarle a depurarlos.

Requisitos previos

Para seguir el procedimiento, necesitará el siguiente hardware:

- Unidad USB: unidad de memoria flash FAT32 USB formateada con al menos 1 GB de almacenamiento, para transferir los archivos de registro desde el dispositivo AWS Panorama.

Para extraer los registros del dispositivo

1. Prepare una unidad USB con una carpeta `managed_logs` dentro de otra carpeta `panorama`.

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

2. Conecte la unidad USB al dispositivo.
3. Apague el dispositivo de AWS Panorama.

4. Encienda el dispositivo de AWS Panorama.
5. El dispositivo copia los registros en el dispositivo. El LED de estado parpadea en azul durante el proceso.
6. Los archivos de registro se pueden encontrar entonces dentro del directorio `managed_logs` con el formato `panorama_device_log_v1_dd_hh_mm.img`

No puede descifrar la imagen de registro usted mismo. Trabaje con el servicio de atención al cliente, un gerente de cuentas técnicas de AWS Panorama o un arquitecto de soluciones para coordinar con el equipo de servicio.

Supervisión de dispositivos y aplicaciones con Amazon CloudWatch

Cuando un dispositivo está en línea, AWS Panorama envía las métricas a Amazon CloudWatch. Puede crear gráficos y cuadros de mando con estas métricas en la CloudWatch consola para supervisar la actividad de los dispositivos y configurar alarmas que le notifiquen cuando los dispositivos se desconecten o las aplicaciones detecten errores.

Para ver las métricas en la consola CloudWatch

1. Abra la página Métricas de la consola de AWS Panorama (`PanoramaDeviceMetrics` espacio de nombres).
2. Elija un esquema de dimensiones.
3. Elija métricas para agregarlas al gráfico.
4. Para elegir una estadística diferente y personalizar el gráfico, utilice las opciones de la pestaña Métricas Gráficas . De forma predeterminada, los gráficos utilizan la estadística `Average` para todas las métricas.

> **Precios**
>
> CloudWatch tiene un nivel Always Free. Más allá del límite del nivel gratuito, CloudWatch cobra por las métricas, los cuadros de mando, las alarmas, los registros y la información. Consulte Precios de CloudWatch para obtener más información.

Para obtener más información al respecto CloudWatch, consulta la Guía del CloudWatch usuario de Amazon.

Secciones

- Uso de métricas de dispositivos
- Uso de métricas de aplicación
- Configuración de alarmas

Uso de métricas de dispositivos

Cuando un dispositivo está en línea, envía las métricas a Amazon CloudWatch. Puede usar estas métricas para supervisar la actividad de los dispositivos y activar una alarma si los dispositivos se desconectan.

- `DeviceActive`: se envía periódicamente cuando el dispositivo está activo.

  Dimensiones: `DeviceId` y `DeviceName`.

Visualice la métrica `DeviceActive` con la estadística `Average`.

Uso de métricas de aplicación

Cuando una aplicación detecta un error, envía las métricas a Amazon CloudWatch. Puede utilizar estas métricas para activar una alarma si una aplicación deja de ejecutarse.

- `ApplicationErrors`: el número de errores de aplicación registrados.

  Dimensiones: `ApplicationInstanceName` y `ApplicationInstanceId`.

Visualice las métricas de la aplicación con la estadística `Sum`.

Configuración de alarmas

Para recibir notificaciones cuando una métrica supere un umbral, cree una alarma. Por ejemplo, puede crear una alarma que envíe una notificación cuando la suma de la métrica `ApplicationErrors` se mantenga en 1 durante 20 minutos.

Para crear una alarma

1. Abre la página de alarmas de CloudWatch la consola Amazon.
2. Elija Crear alarma.
3. Elija Seleccionar métrica y busque una métrica para su dispositivo, por ejemplo, `ApplicationErrors` para `applicationInstance-gk75xmplqbqtenlnmz4ehiu7xa`, `my-application`.
4. Siga las instrucciones para configurar una condición, una acción y un nombre para la alarma.

Para obtener instrucciones detalladas, consulta Crear una CloudWatch alarma en la Guía del CloudWatch usuario de Amazon.

Solución de problemas

En los temas siguientes se proporcionan consejos para la solución de errores y problemas que puedan surgir al utilizar la AWS Panorama consola, el dispositivo o el SDK. Si encuentra un problema que no aparece en esta lista, utilice el botón Proporcionar comentarios de esta página para comunicarlo.

Puede encontrar los registros de su dispositivo en la consola de Amazon CloudWatch Logs. El dispositivo carga los registros del código de la aplicación, del software del dispositivo y AWS IoT los procesa a medida que se generan. Para obtener más información, consulte Visualización de los registros de AWS Panorama.

Aprovisionando

Problema: (macOS) Mi ordenador no reconoce la unidad USB incluida con un adaptador USB-C.

Esto puede ocurrir si conecta la unidad USB a un adaptador USB-C que ya esté conectado al ordenador. Intente desconectar el adaptador y volver a conectarlo con la unidad USB ya conectada.

Problema: el aprovisionamiento falla cuando utilizo mi propia unidad USB.

Problema: el aprovisionamiento falla cuando utilizo el puerto USB 2.0 del dispositivo.

El AWS Panorama dispositivo es compatible con dispositivos de memoria flash USB de entre 1 y 32 GB, pero no todos son compatibles. Se han observado algunos problemas al utilizar el puerto USB 2.0 para el aprovisionamiento. Para obtener resultados uniformes, utilice la unidad USB incluida con el puerto USB 3.0 (junto al puerto HDMI).

En el caso del Lenovo ThinkEdge® SE7 0, el dispositivo no incluye una unidad USB. Utilice una unidad USB 3.0 con al menos 1 GB de almacenamiento.

Configuración del dispositivo

Problema: el dispositivo muestra una pantalla en blanco durante el arranque.

Tras completar la secuencia de arranque inicial, que tarda aproximadamente un minuto, el dispositivo muestra una pantalla en blanco durante un minuto o más mientras carga el modelo e inicia la aplicación. Además, el dispositivo no emite vídeo si se conecta una pantalla después de encenderla.

Problema: el aparato no responde cuando mantengo pulsado el botón de encendido para apagarlo.

El aparato tarda hasta 10 segundos en apagarse de forma segura. Debe mantener pulsado el botón de encendido durante solo 1 segundo para iniciar la secuencia de apagado. Para obtener una lista completa de operaciones con botones, consulte Botones y luces del dispositivo de AWS Panorama.

Problema: necesito generar un nuevo archivo de configuración para cambiar los ajustes o reemplazar un certificado perdido.

AWS Panorama no almacena el certificado del dispositivo ni la configuración de red después de descargarlo, y no puede reutilizar los archivos de configuración. Elimine el dispositivo mediante la AWS Panorama consola y cree uno nuevo con un nuevo archivo de configuración.

Configuración de aplicaciones

Problema: Cuando ejecuto varias aplicaciones, no puedo controlar cuál utiliza la salida HDMI.

Al implementar varias aplicaciones que tienen nodos de salida, la aplicación que se inició más recientemente utiliza la salida HDMI. Si esta aplicación deja de ejecutarse, otra aplicación puede utilizar la salida. Para permitir que solo una aplicación acceda a la salida, elimine el nodo de salida y periferia correspondiente del manifiesto de la aplicación de la otra aplicación y vuelva a implementarla.

Problema: el resultado de la aplicación no aparece en los registros.

Configure un registrador de Python para escribir archivos de registro en `/opt/aws/panorama/logs`. Estos se capturan en un flujo de registro para el nodo contenedor de código. Los flujos de salida y error estándar se capturan en un flujo de registro independiente denominado `console-output`. Si usa `print`, use la opción `flush=True` para evitar que los mensajes se atasquen en el búfer de salida.

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

Fuente: AWS Panorama servicio

Cada vez que implementa un cambio en una aplicación, registra una versión de parche que representa la configuración del paquete y los archivos de activos de cada paquete que utiliza. Utilice el script de limpieza de parches para anular el registro de las versiones de parches no utilizadas.

Transmisiones de cámara

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

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

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

Fuente: registro del nodo de la cámara

El dispositivo no se puede conectar a la transmisión de cámara de la aplicación. Cuando esto ocurre, la salida de vídeo queda en blanco o se bloquea en el último fotograma procesado mientras la aplicación espera un fotograma de vídeo del SDK de la AWS Panorama aplicación. El software del dispositivo intenta conectarse a la transmisión de la cámara y registra los errores de tiempo de espera en el registro del nodo de la cámara. Compruebe que la URL de la transmisión de la cámara es correcta y que el tráfico RTSP se puede enrutar entre la cámara y el dispositivo dentro de la red. Para obtener más información, consulte Conexión del dispositivo de AWS Panorama a su red.

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

Fuente: registro OCC

No se AWS Secrets Manager encuentra el secreto de las credenciales de la transmisión de la cámara. Elimine la transmisión de la cámara y vuelva a crearla.

Error: Camera did not provide an H264 encoded stream

Fuente: registro del nodo de la cámara

La transmisión de la cámara tiene una codificación distinta de la H.264, como la H.265. Vuelva a implementar la aplicación con una transmisión de cámara H.264. Para obtener más información sobre las cámaras compatibles, consulte Cámaras compatibles.

Seguridad en AWS Panorama

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Auditores independientes prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener información sobre los programas de conformidad que se aplican a AWS Panorama, consulte Servicios de AWS en el ámbito del programa de conformidad.
- Seguridad en la nube: su responsabilidad se determina según el servicio de AWS que utiliza. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS Panorama. En los siguientes temas, se le mostrará cómo configurar AWS Panorama para satisfacer sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros servicios de AWS que le ayudarán a monitorear y a proteger los recursos de AWS Panorama.

Temas

Características de seguridad del dispositivo de AWS Panorama

Para proteger sus aplicaciones, modelos y hardware contra códigos maliciosos y otras vulnerabilidades, el dispositivo de AWS Panorama implementa un amplio conjunto de características de seguridad. Entre otras, se incluyen las siguientes.

- Cifrado de disco completo: el dispositivo implementa el cifrado de disco completo de la configuración de claves unificadas de Linux (LUKS2). Todos los datos del software y de las aplicaciones del sistema se cifran con una clave específica del dispositivo. Incluso con acceso físico al dispositivo, un atacante no puede inspeccionar el contenido de su almacenamiento.
- Asignación al azar del diseño de la memoria: para protegerse contra los ataques dirigidos al código ejecutable cargado en la memoria, el dispositivo de AWS Panorama utiliza la asignación al azar del diseño del espacio de direcciones (ASLR). La ASLR asigna la ubicación del código del sistema operativo al azar a medida que se carga en la memoria. Esto evita el uso de exploits que intenten sobrescribir o ejecutar secciones específicas del código al predecir dónde se almacenará durante el tiempo de ejecución.
- Entorno de ejecución confiable: el dispositivo utiliza un entorno de ejecución confiable (TEE) basado en ARM TrustZone, con recursos de almacenamiento, memoria y procesamiento aislados. Solo una aplicación de confianza, que se ejecuta en un sistema operativo independiente dentro del TEE, puede acceder a las claves y otros datos confidenciales almacenados en la zona de confianza. El software del dispositivo de AWS Panorama se ejecuta en un entorno Linux que no es de confianza junto con el código de la aplicación. Solo puede acceder a las operaciones criptográficas realizando una solicitud a la aplicación segura.
- Aprovisionamiento seguro: al aprovisionar un dispositivo, las credenciales (claves, certificados y otro material criptográfico) que transfiera al dispositivo solo son válidas durante un período breve de tiempo. El dispositivo utiliza las credenciales de corta duración para conectarse AWS IoT y solicita un certificado para sí mismo que sea válido durante más tiempo. El servicio AWS Panorama genera credenciales y las cifra con una clave codificada en el dispositivo. Solo el dispositivo que solicitó el certificado puede descifrarlo y comunicarse con AWS Panorama.
- Arranque seguro: cuando el dispositivo se inicia, cada componente de software se autentica antes de ejecutarse. La ROM de arranque, un software codificado en el procesador que no se puede modificar, utiliza una clave de cifrado codificada para descifrar el gestor de arranque, lo que valida el núcleo del entorno de ejecución de confianza, etc.

- Núcleo firmado: los módulos del núcleo se firman con una clave de cifrado asimétrica. El núcleo del sistema operativo descifra la firma con la clave pública y comprueba que coincide con la firma del módulo antes de cargarla en la memoria.
- dm-verity: de forma similar a como se validan los módulos del núcleo, el dispositivo utiliza la función `dm-verity` del mapeador de dispositivos de Linux para comprobar la integridad de la imagen del software del dispositivo antes de montarlo. Si se modifica el software del dispositivo, no se ejecutará.
- Prevención de reversión: al actualizar el software del dispositivo, el dispositivo funde un fusible electrónico en el SoC (sistema integrado en un chip). Cada versión del software prevé que se queme un número cada vez mayor de fusibles y no podrá funcionar si se queman más fusibles.

Prácticas recomendadas de seguridad del dispositivo de AWS Panorama

Tenga en cuenta las siguientes prácticas recomendadas al utilizar el dispositivo de AWS Panorama.

- Asegure físicamente el dispositivo: instale el dispositivo en un bastidor de servidores cerrado o en una sala segura. Limite el acceso físico al dispositivo al personal autorizado.
- Proteja la conexión de red del dispositivo: conecte el dispositivo a un enrutador que limite el acceso a los recursos internos y externos. El dispositivo debe conectarse a las cámaras, que pueden estar en una red interna segura. También necesita conectarse a AWS. Utilice el segundo puerto Ethernet únicamente para la redundancia física y configure el enrutador para que solo permita el tráfico necesario.

  Utilice una de las configuraciones de red recomendadas para planificar el diseño de la red. Para obtener más información, consulte Conexión del dispositivo de AWS Panorama a su red.
- Formatee la unidad USB: después de aprovisionar un dispositivo, extraiga la unidad USB y formatéela. El dispositivo no utiliza la unidad USB después de registrarse en el servicio AWS Panorama. Formatee la unidad para eliminar las credenciales temporales, los archivos de configuración y los registros de aprovisionamiento.
- Mantenga el dispositivo actualizado: aplique las actualizaciones del software del dispositivo de forma oportuna. Cuando ve un dispositivo en la consola de AWS Panorama, la consola le notifica si hay una actualización de software disponible. Para obtener más información, consulte Administración de un dispositivo de AWS Panorama.

  Con la operación de la DescribeDeviceAPI, puede automatizar la búsqueda de actualizaciones comparando los `CurrentSoftware` campos `LatestSoftware` y. Si la última versión del software es diferente de la versión actual, aplique la actualización con la consola o mediante la CreateJobForDevicesoperación.
- Si deja de usar un dispositivo, reinícielo: antes de sacar el dispositivo de su centro de datos seguro, reinícielo por completo. Con el dispositivo apagado y enchufado, pulse el botón de encendido y el botón de reinicio simultáneamente durante 5 segundos. De este modo, se eliminan las credenciales de la cuenta, las aplicaciones y los registros del dispositivo.

  Para obtener más información, consulte Botones y luces del dispositivo de AWS Panorama.
- Limite el acceso a AWS Panorama y a otros servicios de AWS: AWSPanoramaFullAccessproporciona acceso a todas las operaciones de la API de AWS

Panorama y, si es necesario, acceso a otros servicios. Siempre que sea posible, la política limita el acceso a los recursos en función de las convenciones de nomenclatura. Por ejemplo, proporciona acceso a AWS Secrets Manager los secretos cuyos nombres comienzan por`panorama`. Para los usuarios que necesitan acceso de solo lectura o acceso a un conjunto de recursos más específico, usen la política administrada como punto de partida para sus políticas de privilegios mínimos.

Para obtener más información, consulte Políticas de IAM basadas en identidad para AWS Panorama.

Protección de los datos en AWS Panorama

El modelo de se aplica a protección de datos en AWS Panorama. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulta las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulta Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con AWS Panorama u otro tipo de Servicios de AWS uso de la consola AWS CLI, la API o AWS SDKs. Cualquier dato que ingrese en

etiquetas o campos de texto de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Secciones

Cifrado en tránsito

Los puntos de conexión de la API de AWS Panorama solo admiten conexiones seguras a través de HTTPS. Cuando administra recursos de AWS Panorama con la AWS Management Console, el SDK de AWS o la API de AWS Panorama, toda la comunicación se cifra con seguridad de la capa de transporte (TLS). La comunicación entre el dispositivo de AWS Panorama y AWS también se cifra con TLS. La comunicación entre el dispositivo de AWS Panorama y las cámaras a través de RTSP no está cifrada.

Para obtener una lista completa de los puntos de conexión de la API, consulte Regiones de AWS y puntos de conexión en la Referencia general de AWS.

Dispositivo de AWS Panorama

El dispositivo de AWS Panorama tiene puertos físicos para Ethernet, vídeo HDMI y almacenamiento USB. La ranura para tarjetas SD, el wifi y el Bluetooth no se pueden utilizar. El puerto USB solo se utiliza durante el aprovisionamiento para transferir un archivo de configuración al dispositivo.

El contenido del archivo de configuración, que incluye el certificado de aprovisionamiento del dispositivo y la configuración de red, no está cifrado. AWS Panorama no almacena estos archivos; solo se pueden recuperar cuando se registra un dispositivo. Tras transferir el archivo de configuración a un dispositivo, elimínelo del ordenador y del dispositivo de almacenamiento USB.

Todo el sistema de archivos del dispositivo está cifrado. Además, el dispositivo aplica varias protecciones a nivel del sistema, incluida la protección antirretroactiva para las actualizaciones de software necesarias, el núcleo y el gestor de arranque firmados y la verificación de la integridad del software.

Cuando deje de utilizar el dispositivo, realice un restablecimiento completo para eliminar los datos de la aplicación y restablecer el software del dispositivo.

Aplicaciones

Usted controla el código que implementa en su dispositivo. Valide todo el código de la aplicación para detectar problemas de seguridad antes de implementarlo, independientemente de su origen. Si utiliza bibliotecas de terceros en su aplicación, tenga en cuenta detenidamente las políticas de licencia y soporte de esas bibliotecas.

El uso de la CPU, la memoria y el disco de la aplicación no está limitado por el software del dispositivo. Una aplicación que utilice demasiados recursos puede afectar negativamente a otras aplicaciones y al funcionamiento del dispositivo. Pruebe las aplicaciones por separado antes de combinarlas o implementarlas en entornos de producción.

Los activos de las aplicaciones (códigos y modelos) no están aislados del acceso desde su cuenta, dispositivo o entorno de compilación. Las imágenes de contenedores y los archivos de modelos generados por la CLI de la aplicación AWS Panorama no están cifrados. Utilice cuentas independientes para las cargas de trabajo de producción y permita el acceso solo en función de las necesidades.

Otros servicios

Para almacenar sus modelos y contenedores de aplicaciones de forma segura en Amazon S3, AWS Panorama utiliza el cifrado del lado del servidor con una clave que administra Amazon S3. Para obtener más información, consulte Protección de los datos mediante cifrado en la Guía del usuario de Amazon Simple Storage Service.

Las credenciales de Camera Stream se cifran cuando están inactivas AWS Secrets Manager. El rol de IAM del dispositivo le otorga permiso para recuperar el secreto con el fin de acceder al nombre de usuario y la contraseña de la transmisión.

El dispositivo AWS Panorama envía los datos de registro a Amazon CloudWatch Logs. CloudWatch Logs cifra estos datos de forma predeterminada y se puede configurar para usar una clave administrada por el cliente. Para obtener más información, consulte Cifrar datos de registro en CloudWatch Logs utilizando AWS KMS la Guía del usuario de Amazon CloudWatch Logs.

Gestión de identidades y acceso para AWS Panorama

AWS Identity and Access Management (IAM) es una Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. Los administradores de IAM controlan quién puede estar autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de AWS Panorama. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realice en AWS Panorama.

Usuario de servicio: si utiliza el servicio AWS Panorama para realizar su trabajo, su administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de AWS Panorama para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos al administrador. Si no puede acceder a una característica de AWS Panorama, consulte Solución de problemas de identidades y accesos en AWS Panorama .

Administrador de servicio: si está a cargo de los recursos de AWS Panorama en su empresa, probablemente tenga acceso completo a AWS Panorama. Su trabajo consiste en determinar a qué características y recursos de AWS Panorama deben acceder los usuarios del servicio. Luego,

debe enviar solicitudes a su gestionador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con AWS Panorama, consulte Funcionamiento de AWS Panorama con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer más detalles sobre cómo escribir políticas para administrar el acceso a AWS Panorama. Para consultar ejemplos de políticas basadas en identidades de AWS Panorama que puede utilizar en IAM, consulte Ejemplos de políticas de AWS Panorama basadas en identidades.

Autenticación con identidades

La autenticación es la forma de iniciar sesión para AWS usar sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su gestionador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre la firma de solicitudes, consulte AWS Signature Versión 4 para solicitudes API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Autenticación multifactor AWS en IAM en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utiliza el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulta Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puedes iniciar sesión como grupo. Puedes usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una persona determinada. Para asumir temporalmente un rol de IAM en el AWS Management Console, puede cambiar de un rol de usuario

a uno de IAM (consola). Puedes asumir un rol llamando a una operación de AWS API AWS CLI o usando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulta Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puedes crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles de federación, consulte Crear un rol para un proveedor de identidad de terceros (federación) en la Guía de usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué puedes acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulta Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puedes asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puedes utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulta Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en AWS ellas, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para

obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulta Reenviar sesiones de acceso.

- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puedes asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y realizan AWS CLI solicitudes a la AWS API. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulta Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puedes crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utiliza para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puedes asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades puedes clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para obtener más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios puedes utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puedes realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas puedes establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puedes conceder a una entidad de IAM (usuario o rol de IAM). Puedes establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulta Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): SCPs son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre Organizations SCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de control de recursos (RCPs): RCPs son políticas de JSON que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las políticas de IAM asociadas a cada recurso que poseas. El RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades, incluidos los permisos Usuario raíz de la cuenta de AWS, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de Servicios de AWS ese apoyo RCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades

del rol y las políticas de la sesión. Los permisos también puedes proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulta Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Funcionamiento de AWS Panorama con IAM

Antes de utilizar IAM para administrar el acceso a AWS Panorama, debe comprender qué características de IAM están disponibles para su uso con AWS Panorama. Para obtener una visión general de cómo AWS Panorama y otros AWS servicios funcionan con IAM, consulte AWS los servicios que funcionan con IAM en la Guía del usuario de IAM.

Para obtener información general acerca de los permisos, las políticas y roles a medida que AWS Panorama utiliza, consulte AWS Panorama permisos.

Ejemplos de políticas de AWS Panorama basadas en identidades

De forma predeterminada, los usuarios y los roles de IAM no tienen permiso para crear, ver ni modificar recursos de AWS Panorama. Tampoco pueden realizar tareas con la AWS Management Console AWS CLI, o la API. AWS Un administrador de IAM debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos especificados que necesiten. El administrador debe adjuntar esas políticas a los usuarios o grupos de IAM que necesiten esos permisos.

Para obtener más información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas de JSON de ejemplo, consulte Creación de políticas en la pestaña JSON en la Guía del usuario de IAM.

Temas

- Prácticas recomendadas relativas a políticas
- Uso de la consola de AWS Panorama
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas relativas a políticas

Las políticas basadas en identidades determinan si alguien puede crear, acceder o eliminar los recursos de AWS Panorama en su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos en muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se puedes llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulta Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utiliza condiciones en las políticas de IAM para restringir aún más el acceso: puedes agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puedes escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la

MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola de AWS Panorama

Para acceder a la consola de AWS Panorama, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de AWS Panorama de su AWS cuenta. Si crea una política basada en identidad que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles de IAM) que tengan esa política.

Para obtener más información, consulte Políticas de IAM basadas en identidad para AWS Panorama

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas gestionadas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o 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 políticas administradas para AWS Panorama

Una política AWS gestionada es una política independiente creada y administrada por. AWS AWS Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

AWS Panorama dispone de las siguientes políticas administradas. Para ver el contenido completo y el historial de cambios de cada política, consulte las páginas enlazadas en la consola de IAM.

- AWSPanoramaFullAccess— Proporciona acceso completo a AWS Panorama, a los puntos de acceso de AWS Panorama en Amazon S3, a las credenciales de los dispositivos y a los registros

de los dispositivos en Amazon CloudWatch. AWS Secrets Manager Incluye permiso para crear un rol vinculado a un servicio para AWS Panorama.

- AWSPanoramaServiceLinkedRolePolicy— Permite a AWS Panorama gestionar los recursos en AWS IoT, AWS Secrets Manager y AWS Panorama.
- AWSPanoramaApplianceServiceRolePolicy— Permite a un dispositivo AWS Panorama cargar registros y obtener objetos de los puntos de acceso de Amazon S3 creados por AWS Panorama. CloudWatch

Actualizaciones de AWS Panorama de las políticas AWS administradas

En la siguiente tabla se describen las actualizaciones de las políticas administradas para AWS Panorama.

Cambio	Descripción	Fecha
AWSPanoramaApplianceServiceRolePolicy — Actualización de una política existente	Sustituya StringLike la condición por « ArnLike para escribir ARNs».	2024-12-10
AWSPanoramaFullAccess — Actualización de una política existente	Sustituya StringLike la condición por « ArnLike para escribir ARNs».	2024-12-10
AWSPanoramaFullAccess — Actualización de una política existente	Se agregaron permisos a la política de usuario para permitir a los usuarios ver los grupos de registros en la consola de CloudWatch registros.	13/01/2022
AWSPanoramaFullAccess — Actualización de una política existente	Se agregaron permisos a la política de usuario para permitir a los usuarios administrar el rol vinculado al servicio AWS Panorama y acceder a los recursos	20 de octubre de 2021

Cambio	Descripción	Fecha
	de AWS Panorama en otros servicios, como IAM CloudWatch, Amazon S3 y Secrets Manager.	
AWSPanoramaApplianceServiceRolePolicy — Nueva política	Nueva política para el rol de servicio del dispositivo de AWS Panorama	2021-10-20
AWSPanoramaServiceLinkedRolePolicy — Nueva política	Nueva política para el rol vinculado a servicios de AWS Panorama.	2021-10-20
AWS Panorama comenzó a realizar un seguimiento de los cambios	AWS Panorama comenzó a realizar un seguimiento de los cambios en sus políticas AWS administradas.	20 de octubre de 2021

Uso de roles vinculados a servicios para AWS Panorama

AWS Panorama usa roles vinculados al AWS Identity and Access Management servicio (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM al que se vincula directamente. AWS Panorama Los roles vinculados al servicio están predefinidos AWS Panorama e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

Un rol vinculado a un servicio facilita la configuración AWS Panorama , ya que no es necesario añadir manualmente los permisos necesarios. AWS Panorama define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo AWS Panorama puede asumir sus funciones. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo puede eliminar un rol vinculado a servicios después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de AWS Panorama , ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados al servicio, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la

columna Service-linked role (Rol vinculado al servicio). Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Secciones

- Permisos de roles vinculados al servicio para AWS Panorama
- Crear un rol vinculado a un servicio para AWS Panorama
- Edición de un rol vinculado a un servicio para AWS Panorama
- Eliminar un rol vinculado a un servicio para AWS Panorama
- Regiones compatibles para los roles vinculados al servicio AWS Panorama

Permisos de roles vinculados al servicio para AWS Panorama

AWS Panorama utiliza el rol vinculado al servicio denominado AWSServiceRoleForAWSPanorama: Permite que AWS Panorama gestione los recursos en AWS IoT, AWS Secrets Manager y AWS Panorama.

El rol AWSService RoleFor AWSPanorama vinculado al servicio confía en los siguientes servicios para asumir el rol:

- `panorama.amazonaws.com`

La política de permisos del rol permite AWS Panorama realizar las siguientes acciones:

- Supervise AWS Panorama los recursos
- Administre AWS IoT los recursos del AWS Panorama dispositivo
- Acceda a AWS Secrets Manager los secretos para obtener las credenciales de la cámara

Para ver una lista completa de permisos, consulta la AWSPanorama ServiceLinkedRolePolicy política en la consola de IAM.

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Crear un rol vinculado a un servicio para AWS Panorama

No necesita crear manualmente un rol vinculado a servicios. Al registrar un dispositivo en la AWS Management Console, la o la AWS API AWS CLI, se AWS Panorama crea automáticamente el rol vinculado al servicio.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al registrar un dispositivo, vuelve a AWS Panorama crear el rol vinculado al servicio para usted.

Edición de un rol vinculado a un servicio para AWS Panorama

AWS Panorama no permite editar el rol vinculado al AWSService RoleFor AWSPanorama servicio. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminar un rol vinculado a un servicio para AWS Panorama

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Así no tendrá una entidad no utilizada que no se supervise ni mantenga de forma activa. Sin embargo, debe limpiar los recursos de su rol vinculado al servicio antes de eliminarlo manualmente.

Para eliminar los AWS Panorama recursos que utiliza AWSService RoleForAWSPanorama, utilice los procedimientos de las siguientes secciones de esta guía.

- Eliminar versiones y aplicaciones
- Anulación del registro de un dispositivo

> **Note**
>
> Si el AWS Panorama servicio utiliza el rol al intentar eliminar los recursos, es posible que la eliminación no se realice correctamente. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar el rol AWSService RoleFor AWSPanorama vinculado al servicio, usa la consola de IAM AWS CLI, la o la API. AWS Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones compatibles para los roles vinculados al servicio AWS Panorama

AWS Panorama admite el uso de funciones vinculadas al servicio en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte Puntos de conexión y Regiones de AWS.

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar un confuso problema de diputado. La suplantación entre servicios puedes producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puedes manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger los datos de todos los servicios cuyos directores de servicio tengan acceso a los recursos de su cuenta.

Recomendamos utilizar las claves de contexto de condición `aws:SourceAccount`global `aws:SourceArn`y las claves contextuales en las políticas de recursos para limitar los permisos que se AWS Panorama otorgan a otro servicio al recurso. Si se utilizan ambas claves contextuales de condición global, el valor `aws:SourceAccount` y la cuenta del valor `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utilicen en la misma declaración de política.

El valor de `aws:SourceArn` debe ser el ARN de un AWS Panorama dispositivo.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si especifica varios recursos, utiliza la clave de condición de contexto global `aws:SourceArn` con comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

Para obtener instrucciones sobre cómo proteger la función de servicio que se AWS Panorama utiliza para conceder permisos al AWS Panorama dispositivo, consulteAsegurar el rol del dispositivo.

Solución de problemas de identidades y accesos en AWS Panorama

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que puedan surgir cuando trabaje con AWS Panorama e IAM.

Temas

No tengo autorización para realizar una acción en AWS Panorama

Si AWS Management Console le indica que no está autorizado a realizar una acción, debe ponerse en contacto con el administrador para obtener ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para ver detalles sobre una aplicación, pero no tiene permisos `panorama:DescribeAppliance`.

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

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso `my-appliance` mediante la acción `panorama:DescribeAppliance`.

No estoy autorizado a realizar lo siguiente: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, se deben actualizar las políticas a fin de permitirle pasar un rol a AWS Panorama.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en AWS Panorama. Sin embargo, la acción

requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir a personas externas a mi cuenta de AWS el acceso a mis recursos de AWS Panorama

Puedes crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puedes especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admiten políticas basadas en recursos o listas de control de acceso (ACLs), puede utilizar esas políticas para permitir que las personas accedan a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para obtener información acerca de si AWS Panorama admite estas características, consulte Funcionamiento de AWS Panorama con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro de su propiedad en la Cuenta de AWS Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulta Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Validación de la conformidad en AWS Panorama

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Cumplimiento de seguridad y gobernanza: en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- Referencia de servicios válidos de HIPAA: muestra una lista con los servicios válidos de HIPAA. No todos Servicios de AWS cumplen con los requisitos de la HIPAA.
- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulta la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar

actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.

- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Consideraciones adicionales sobre la presencia de personas

A continuación, se muestran algunas prácticas recomendadas que se deben tener en cuenta al utilizar AWS Panorama en situaciones en las que podrían estar presentes personas:

- Asegúrese de conocer y cumplir con todas las leyes y reglamentos aplicables a su caso de uso. Esto puede incluir las leyes relacionadas con la posición y el campo de visión de las cámaras, los requisitos de aviso y señalización a la hora de colocar y utilizar las cámaras y los derechos de las personas que puedan estar presentes en tus vídeos, incluidos sus derechos de privacidad.
- Tenga en cuenta el efecto de sus cámaras en las personas y en su privacidad. Además de cumplir con los requisitos legales, considere si sería apropiado colocar un aviso en las áreas donde se encuentran sus cámaras y si las cámaras deben colocarse a plena vista y sin obstrucciones, de modo que las personas no se sorprendan de que estén frente a las cámaras.
- Establezca políticas y procedimientos adecuados para el funcionamiento de sus cámaras y revise los datos obtenidos de las cámaras.
- Tenga en cuenta los controles de acceso, las limitaciones de uso y los períodos de retención adecuados para los datos obtenidos de sus cámaras.

Seguridad de la infraestructura en AWS Panorama

Como servicio gestionado, AWS Panorama está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a AWS Panorama a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Implementación del dispositivo de AWS Panorama en su centro de datos

El dispositivo AWS Panorama necesita acceso a Internet para comunicarse con AWS los servicios. También necesita acceso a su red interna de cámaras. Es importante considerar detenidamente la configuración de la red y proporcionar a cada dispositivo únicamente el acceso que necesita. Tenga cuidado si su configuración permite que el dispositivo de AWS Panorama actúe como puente a una red de cámaras IP de confianza.

Usted es responsable de lo siguiente:

- La seguridad de red física y lógica del dispositivo de AWS Panorama.
- Utilizar de forma segura las cámaras conectadas a la red cuando utilice el dispositivo de AWS Panorama.
- Mantener actualizados el dispositivo de AWS Panorama y el software de la cámara.
- Cumplir con cualquier ley o reglamento aplicable relacionado con el contenido de los vídeos e imágenes que recopile en sus entornos de producción, incluidos los relacionados con la privacidad.

El dispositivo de AWS Panorama utiliza transmisiones de cámara RTSP sin cifrar. Para obtener más información sobre cómo conectar el dispositivo de AWS Panorama a la red, consulte Conexión del dispositivo de AWS Panorama a su red. Para obtener más información sobre el cifrado, consulte Protección de los datos en AWS Panorama.

Software de entorno de tiempo de ejecución en AWS Panorama

AWS Panorama proporciona software que ejecuta el código de su aplicación en un entorno basado en Ubuntu Linux en el dispositivo de AWS Panorama. AWS Panorama es responsable de mantener actualizado el software de la imagen del dispositivo. AWS Panorama publica periódicamente actualizaciones de software, que puede aplicar mediante la consola de AWS Panorama.

Puede utilizar las bibliotecas del código de su aplicación instalándolas en el código de la aplicación `Dockerfile`. Para garantizar la estabilidad de las aplicaciones en todas las compilaciones, elija una versión específica de cada biblioteca. Actualice sus dependencias con regularidad para solucionar los problemas de seguridad.

Versiones

En la siguiente tabla se muestra cuándo se publicaron las funciones y las actualizaciones de software para el AWS Panorama servicio, el software y la documentación. Para garantizar el acceso a todas las funciones, actualice el AWS Panorama dispositivo a la versión de software más reciente. Para más información sobre una versión, consulte el tema enlazado.

Cambio	Descripción	Fecha
Notificación del fin del soporte	Aviso de fin del soporte: el 31 de mayo de 2026, AWS finalizará el soporte para AWS Panorama. Después del 31 de mayo de 2026, ya no podrás acceder a la AWS Panorama consola ni a AWS Panorama los recursos. Para obtener más información, consulta AWS Panorama el fin del soporte.	20 de mayo de 2025
Actualización de las políticas administradas	AWS Identity and Access Management se AWS Panorama han actualizado las políticas gestionadas para. Para obtener más información, consulte las Políticas administradas por AWS.	10 de diciembre de 2024
Actualización del software del dispositivo	La versión 7.0.13 es una actualización de versión importante que cambia la forma en que el dispositivo gestiona las actualizaciones de software. Si restringe la comunicación de red saliente desde el dispositivo o lo	28 de diciembre de 2023

	conecta a una subred de VPC privada, debe permitir el acceso a puntos de conexión y puertos adicionales antes de aplicar la actualización. Para más información, consulte el registro de cambios.	
Actualización del software del dispositivo	La versión 6.2.1 incluye correcciones de errores. Para más información, consulte el registro de cambios.	6 de septiembre de 2023
Actualización del software del dispositivo	La versión 6.0.8 incluye correcciones de errores y mejoras de seguridad. Para más información, consulte el registro de cambios.	6 de julio de 2023
Actualización del software del dispositivo	La versión 5.1.7 incluye correcciones de errores y mejoras en la gestión de errores. Para más información, consulte el registro de cambios.	31 de marzo de 2023
Actualizaciones de consola	Ahora puede comprar el AWS Panorama dispositivo desde la consola de administración. Para conceder permiso a un usuario para comprar dispositivos, consulte Políticas de IAM basadas en la identidad para AWS Panorama.	2 de febrero de 2023

Actualización del software del dispositivo	La versión 5.0.74 incluye correcciones de errores y mejoras en la gestión de errores. Para más información, consulte el registro de cambios.	23 de enero de 2023
Actualización de la API	Se agregó la opción de `AllowMajorVersionUpdate` a `OTAJobConfig` para hacer que las actualizaciones de las versiones principales del software del dispositivo sean opcionales. Para obtener más información, consulte CreateJobForDevices.	19 de enero de 2023
Nueva herramienta para desarrolladores	Una nueva herramienta, la «carga lateral», está disponible en el GitHub repositorio de AWS Panorama muestras. Puede usar esta herramienta para actualizar el código de la aplicación sin crear ni implementar un contenedor. Para obtener más información, consulte el README.	16 de noviembre de 2022

Actualización de la imagen base de la aplicación	La versión 1.2.0 añade una opción de tiempo de espera a `video_in.get()` , establece la variable de entorno de `AWS_REGION` y mejora la gestión de errores. Para más información, consulte el registro de cambios.	16 de noviembre de 2022
Actualización del software del dispositivo	La versión 5.0.42 incluye correcciones de errores y actualizaciones de seguridad . Para más información, consulte el registro de cambios.	16 de noviembre de 2022
Actualización del software del dispositivo	La versión 5.0.7 añade soporte para reiniciar los dispositivos de forma remota y pausar las transmisiones de la cámara de forma remota. Para más información, consulte el registro de cambios.	13 de octubre de 2022
Actualización del software del dispositivo	La versión 4.3.93 añade soporte para recuperar registros de un dispositivo sin conexión. Para más información, consulte el registro de cambios.	24 de agosto de 2022

Actualización del software del dispositivo	La versión 4.3.72 incluye correcciones de errores y actualizaciones de seguridad . Para más información, consulte el registro de cambios.	23 de junio de 2022
AWS PrivateLink apoyo	AWS Panorama admite puntos finales de VPC para administrar los AWS Panorama recursos de una subred privada. Para obtener más información, consulte Uso de puntos de conexión de VPC.	2 de junio de 2022
Actualización del software del dispositivo	La versión 4.3.55 mejora la utilización del almacenamiento del registro de `console_output` . Para más información, consulte el registro de cambios.	5 de mayo de 2022
Lenovo® 0 ThinkEdge SE7	Lenovo AWS Panorama tiene disponible un nuevo electrodoméstico para. El Lenovo ThinkEdge® SE7 0, equipado con la tecnología Nvidia Jetson Xavier NX, admite las mismas funciones que el AWS Panorama dispositivo. Para más información, consulte Dispositivos compatibles.	6 de abril de 2022

Actualización de la imagen base de la aplicación	La versión 1.1.0 mejora el rendimiento al ejecutar subprocesos en segundo plano y añade un indicador (is_cached) a los objetos multimedia que indica si la imagen está actualizada. Para obtener más información, consulte gallery.ecr.aws.	29 de marzo de 2022
Actualización del software del dispositivo	La versión 4.3.45 añade compatibilidad con el acceso a la GPU y los puertos de entrada. Para más informaci ón, consulte el registro de cambios.	24 de marzo de 2022
Actualización del software del dispositivo	La versión 4.3.35 mejora la seguridad y el rendimien to. Para más informaci ón, consulte el registro de cambios.	22 de febrero de 2022
Actualización de las políticas administradas	AWS Identity and Access Management se han actualiza do las políticas AWS Panorama gestionadas para. Para obtener más información, consulte las Políticas administr adas por AWS.	13 de enero de 2022

Registros de aprovisionamiento	Con el software del dispositivo 4.3.23, el dispositivo escribe los registros en una unidad USB durante el aprovisionamiento. Para obtener más información, consulte Registros.	13 de enero de 2022
Configuración del servidor de NTP	Ahora puede configurar el AWS Panorama dispositivo para que utilice un servidor NTP específico para la sincronización del reloj. Configure los ajustes de NTP durante la configuración del dispositivo con otros ajustes de red. Para más información, consulte Configuración.	13 de enero de 2022
Regiones adicionales	AWS Panorama ya está disponible en las regiones de Asia Pacífico (Singapur) y Asia Pacífico (Sídney).	13 de enero de 2022
Actualización del software del dispositivo	La versión 4.3.4 añade compatibilidad con la configuración de `precisionMode` de los modelos y actualiza el comportamiento de registro. Para más información, consulte el registro de cambios.	8 de noviembre de 2021

Actualización de las políticas administradas	AWS Identity and Access Management se AWS Panorama han actualizado las políticas gestionadas para. Para obtener más información, consulte las Políticas administradas por AWS.	20 de octubre de 2021
Disponibilidad general	AWS Panorama ahora está disponible para todos los clientes de las regiones EE.UU. Este (Norte de Virginia), EE.UU. Oeste (Oregón), Europa (Irlanda) y Canadá (Centro). Para comprar un AWS Panorama electrodoméstico, visite AWS Panorama.	20 de octubre de 2021
Vista previa	AWS Panorama está disponible mediante invitación en las regiones EE.UU. Este (Norte de Virginia) y EE.UU. Oeste (Oregón).	1 de diciembre de 2020