Guia do Desenvolvedor

AWS Panorama

AWS Panorama: Guia do Desenvolvedor

Table of Contents

Aviso de fim do suporte: em 31 de maio de 2026, AWS encerrará o suporte para AWS Panorama. Depois de 31 de maio de 2026, você não poderá mais acessar o AWS Panorama console ou os AWS Panorama recursos. Para obter mais informações, consulte AWS Panorama Fim do suporte.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.

O que é AWS Panorama?

AWS Panorama é um serviço que traz visão computacional para sua rede de câmeras local. Você instala o AWS Panorama equipamento ou outro dispositivo compatível em seu datacenter, o registra e implanta aplicativos de AWS Panorama visão computacional a partir da nuvem. AWS Panorama funciona com suas câmeras de rede de protocolo de streaming em tempo real (RTSP) existentes. O equipamento executa aplicativos seguros de visão computacional de AWS parceiros ou aplicativos que você mesmo cria com o SDK do AWS Panorama aplicativo.

O AWS Panorama equipamento é um dispositivo de ponta compacto que usa um poderoso system-on-module (SOM) otimizado para cargas de trabalho de aprendizado de máquina. O dispositivo pode executar vários modelos de visão computacional paralelamente em vários streams de vídeo e gerar os resultados em tempo real. Ele foi projetado para uso em ambientes comerciais e industriais e tem proteção contra poeira e líquidos (IP-62).

O AWS Panorama dispositivo permite que você execute aplicativos de visão computacional independentes na borda, sem enviar imagens para a nuvem da AWS. Ao usar o SDK da AWS, você pode se integrar a outros serviços da AWS e usá-los para rastrear dados da aplicação ao longo do tempo. Ao se integrar com outros serviços da AWS, você pode usar AWS Panorama para fazer o seguinte:

- Analisar padrões de tráfego: use o SDK da AWS para registrar dados de análise de varejo no Amazon DynamoDB. Use uma aplicação com tecnologia sem servidor para analisar os dados coletados ao longo do tempo, detectar anomalias nos dados e prever o comportamento futuro.
- Receber alertas de segurança do local: monitore áreas proibidas em uma instalação industrial. Quando a aplicação detectar uma situação potencialmente insegura, faça o upload de uma imagem no Amazon Simple Storage Service (Amazon S3) e envie uma notificação para um tópico do Amazon Simple Notification Service (Amazon SNS) para que os destinatários possam tomar medidas corretivas.
- Melhorar o controle de qualidade: monitore a produção de uma linha de montagem para identificar peças que não estejam em conformidade com os requisitos. Destaque imagens de peças não conformes usando texto e uma caixa delimitadora e exiba-as em um monitor, para que sua equipe de controle de qualidade as analise.
- Coletar dados de treinamento e teste: faça upload de imagens de objetos que seu modelo de visão computacional não conseguiu identificar ou nos quais a confiança na adivinhação do modelo estava no limite. Use uma aplicação com tecnologia sem servidor para criar uma fila de imagens

que precisam ser marcadas. Marque as imagens e use-as para retreinar o modelo na Amazon SageMaker AI.

AWS Panorama usa outros serviços da AWS para gerenciar o AWS Panorama dispositivo, acessar modelos e códigos e implantar aplicativos. AWS Panorama faz o máximo possível sem exigir que você interaja com outros serviços, mas o conhecimento dos serviços a seguir pode ajudá-lo a entender como AWS Panorama funciona.

- SageMaker IA — Você pode usar a SageMaker IA para coletar dados de treinamento de câmeras ou sensores, criar um modelo de aprendizado de máquina e treiná-lo para visão computacional. AWS Panorama usa o SageMaker AI Neo para otimizar modelos para execução no AWS Panorama dispositivo.
- Amazon S3 — Você usa os pontos de acesso do Amazon S3 para preparar o código, os modelos e os arquivos de configuração do aplicativo para implantação em um dispositivo. AWS Panorama
- AWS IoT— AWS Panorama usa AWS IoT serviços para monitorar o estado do AWS Panorama equipamento, gerenciar atualizações de software e implantar aplicativos. Você não precisa usar AWS IoT diretamente.

Para começar a usar o AWS Panorama equipamento e saber mais sobre o serviço, continue Começando com AWS Panorama em.

Fim do suporte do AWS Panorama

Após uma análise cuidadosa, decidimos encerrar o suporte para o Panorama da AWS, a partir de 31 de maio de 2026. O AWS Panorama não aceitará mais novos clientes a partir de 20 de maio de 2025. Como cliente atual com uma conta cadastrada no serviço antes de 20 de maio de 2025, você pode continuar usando os recursos do Panorama da AWS. Depois de 31 de maio de 2026, você não poderá mais usar o AWS Panorama.

Alternativas ao AWS Panorama

Se você estiver interessado em uma alternativa ao AWS Panorama, AWS tem opções para compradores e construtores.

Para uma out-of-the-box solução, a Rede de Parceiros da AWS oferece soluções de vários parceiros. Você pode procurar soluções na Biblioteca de Soluções da AWS de muitos de nossos parceiros.

Essas soluções de parceiros incluem opções para hardware, software, aplicativos de software como serviço (SaaS), soluções gerenciadas ou implementações personalizadas com base em suas necessidades. Essa abordagem fornece uma solução que aborda seu caso de uso sem exigir que você tenha experiência em visão computacional, IA ou desenvolvimento de aplicativos. Isso normalmente proporciona um tempo mais rápido de geração de valor, aproveitando a experiência especializada dos parceiros da AWS.

Se você preferir criar sua própria solução, AWS oferece ferramentas e serviços de IA para ajudá-lo a desenvolver um aplicativo de visão computacional baseado em IA e gerenciar os aplicativos e dispositivos na borda. SageMakerA Amazon fornece um conjunto de ferramentas para criar, treinar e implantar modelos de ML para seu caso de uso com infraestrutura, ferramentas e fluxos de trabalho totalmente gerenciados. Além de permitir que você crie seus próprios modelos, a Amazon SageMaker JumpStart oferece algoritmos de visão computacional integrados que podem ser ajustados ao seu caso de uso específico.

Para gerenciar dispositivos e aplicativos na borda, o AWS IoT Greengrass é uma solução comprovada e segura para implantar e atualizar aplicativos para dispositivos de IoT. Para uma implementação baseada em servidor, o AWS Systems Manager fornece um conjunto de ferramentas para gerenciar servidores e o Amazon EKS Anywhere ou o ECS Anywhere podem gerenciar contêineres de aplicativos em servidores de borda. A Amazon fornece algumas diretrizes para o gerenciamento de dispositivos de ponta, junto com recursos adicionais na Seção 4 do whitepaper Protegendo a Internet das Coisas (IoT) com a AWS. Essa abordagem de criação fornece as ferramentas para acelerar seu desenvolvimento de IA e gerenciamento de dispositivos, ao mesmo tempo em que oferece flexibilidade total para criar uma solução que atenda exatamente aos seus requisitos e se integre à sua infraestrutura de hardware e software existente. Isso normalmente fornece custos operacionais mais baixos para uma solução.

Migração do AWS Panorama

Para mover um aplicativo existente do AWS Panorama para uma implementação alternativa, você precisará substituir o dispositivo de hardware existente, migrar o aplicativo do serviço AWS Panorama e implementar gerenciamento e segurança de borda para a nova solução. Cada uma dessas áreas será explorada em detalhes abaixo:

Substituição de hardware

O dispositivo AWS Panorama existente é baseado na plataforma Nvidia Jetson Xavier. O hardware pode ser substituído por um off-the-shelf dispositivo similar baseado na plataforma Nvidia Jetson

da geração atual que atenda aos seus requisitos ou em um servidor de ponta. Embora a maioria das implantações do AWS Panorama possa ser substituída por um dispositivo similar, vimos alguns clientes que utilizam um grande número de câmeras em um único local descobrirem que um servidor é a melhor alternativa.

Migração de aplicativos

Os aplicativos do AWS Panorama precisam ser reescritos para eliminar o uso de qualquer chamada de API específica do AWS Panorama. Os aplicativos do AWS Panorama oferecem suporte somente à entrada de vídeo por meio de feeds do Real-Time Streaming Protocol (RTSP) usando H.264, e essas entradas de vídeo são fornecidas usando nós de câmera no SDK do dispositivo AWS Panorama.

Para portar um aplicativo existente, você precisará implementar uma classe de aplicativo semelhante ao AWS Panorama para que o código existente possa ser reutilizado principalmente. O código de amostra está disponível no arquivo banner-code.zip que mostra um exemplo dessa implementação usando PyAV e OpenCV.

Essa é uma abordagem simples com uma quantidade mínima de alterações no código, mas tem muitas das mesmas limitações da implementação atual baseada no Panorama AWS em termos dos tipos de streams de vídeo suportados.

Outra opção seria rearquitetar o aplicativo para fazer melhor uso dos recursos do sistema e oferecer suporte a novos recursos do aplicativo. Para essa opção, você usa GStreamerou DeepStreamimplementa o pipeline de mídia da fonte de mídia até os resultados de inferência e a lógica de negócios, ou usa uma implementação de tempo de execução de aprendizado de máquina (ML) mais rica em recursos e com melhor desempenho, como o servidor de inferência Nvidia Triton. Essa abordagem requer mudanças em mais partes do pipeline de processamento de vídeo, mas é mais eficiente e permite mais flexibilidade para suportar uma variedade maior de codecs, tipos de câmeras e outros sensores.

Gerenciamento e segurança de borda

Independentemente do pipeline de mídia, você também precisará implementar um armazenamento seguro para credenciais, por exemplo, nome de usuário e senha do stream RTSP. AWS fornece maneiras diferentes de armazenar parâmetros de forma segura para aplicativos:

- O serviço AWS IoT Device Shadow é usado para armazenar parâmetros que são passados aos aplicativos, bem como para rastrear o status dos aplicativos no dispositivo de borda.

- O AWS Secrets Manager é usado para armazenar essas credenciais para melhor proteger as credenciais de acesso aos fluxos de mídia.
- Se você usa o Amazon EKS ou o Amazon ECS, também pode usar o armazenamento seguro de parâmetros do AWS System Manager para credenciais e outros parâmetros do aplicativo.

A escolha depende dos requisitos de segurança do aplicativo, bem como de quais outros AWS produtos você planeja usar para implementar seu aplicativo.

Ao substituir o dispositivo AWS Panorama por um dispositivo de borda genérico, você também deve implementar os recursos de segurança necessários para seus aplicativos e configurar os dispositivos para que estejam em conformidade com seus requisitos de segurança. AWS fornece orientação sobre isso no pilar de segurança do AWS Well-Architected Framework. Embora a estrutura se concentre principalmente em aplicativos em nuvem, a maioria dos princípios também se aplica aos dispositivos de ponta. Além disso, você deve usar os recursos de segurança de hardware da solução escolhida, como a integração de segurança de hardware do AWS IoT Greengrass V2, e usar os recursos de segurança fornecidos pelo sistema operacional e/ou dispositivo escolhido, como criptografia de disco completo.

Resumo

Embora o AWS Panorama esteja planejando ser encerrado em 31 de maio de 2026, AWS oferece um poderoso conjunto de serviços e soluções de IA/ML na forma de SageMaker ferramentas da Amazon para criar modelos de visão computacional e serviços de gerenciamento de dispositivos, como AWS IoT Greengrass, Amazon EKS e Amazon ECS Anywhere e AWS System Manager para apoiar o desenvolvimento de soluções semelhantes. AWS também tem uma variedade de ofertas de parceiros da Rede de Parceiros da AWS, se você preferir comprar em vez de criar uma solução. Um exemplo de código e orientação de implementação é fornecido para ajudar a migrar para uma solução alternativa, se você preferir. Você deve explorar essas opções para determinar o que funciona melhor para suas necessidades específicas.

Para obter mais detalhes, consulte os seguintes recursos:

- Guia SageMaker do desenvolvedor da Amazon — Documentação detalhada sobre como criar um modelo ou trabalhar com algoritmos de visão computacional integrados disponíveis em SageMaker JumpStart.
- Guia do desenvolvedor do AWS IoT Core — Documentação detalhada sobre como conectar e gerenciar dispositivos de IoT.

- Guia do desenvolvedor do AWS IoT Greengrass V2 — Documentação detalhada sobre como criar, implantar e gerenciar aplicativos de IoT em seus dispositivos.
- Guia do desenvolvedor do ECS Anywhere - Documentação detalhada sobre como executar o ECS na borda.
- Guia de melhores práticas do EKS Anywhere - Documentação detalhada sobre como executar o EKS na borda.
- Biblioteca de soluções da AWS — ofertas de parceiros de uma variedade de fornecedores que oferecem soluções de visão computacional pré-criadas ou personalizadas.
- Panorama FAQs - Informações adicionais sobre o Panorama.

Perguntas frequentes

Qual é o momento para a descontinuação do Panorama?

O anúncio foi feito em 20 de maio de 2025. Após essa data, os clientes que não estiverem ativos no serviço não terão mais acesso ao Panorama. Os clientes ativos poderão continuar usando o serviço normalmente até 31 de maio de 2026. Até esse momento, os clientes precisam mover seus aplicativos para uma solução alternativa e migrar os aplicativos do Panorama. Depois de 31 de maio de 2026, qualquer aplicativo que tente acessar o serviço Panorama não funcionará mais e os dispositivos Panorama não funcionarão mais.

Como os clientes existentes serão afetados?

Os clientes existentes podem continuar usando o serviço normalmente até 31 de maio de 2026. Depois disso, os aplicativos que tentarem acessar o Panorama não funcionarão mais. Os dispositivos Panorama também não funcionarão mais após essa data.

Novos clientes estão sendo aceitos?

Não. A partir de 20 de maio de 2025, somente clientes que sejam usuários ativos do Panorama terão acesso ao serviço. Se um cliente tiver aplicativos no serviço de uso anterior que ele precise acessar, ele poderá criar um caso com o suporte ao cliente para solicitar acesso à sua conta. Se um cliente não tiver usado o serviço anteriormente, ele não terá acesso.

Quais são as alternativas que os clientes podem explorar?

AWS oferece uma variedade de serviços que podem substituir os recursos do Panorama. Recomendamos que os clientes utilizem off-the-shelf hardware e gerenciem o dispositivo e o

aplicativo por meio da combinação do AWS IoT Core, do AWS IoT Greengrass, do Amazon AKS Anywhere, do Amazon ECS Anywhere e/ou do AWS System Manager que atenda aos seus requisitos. A rede de parceiros da AWS também tem várias soluções disponíveis de parceiros com experiência específica em visão computacional que os clientes podem considerar.

Como os clientes podem migrar do Panorama?

Os aplicativos Panorama precisam ser modificados para remover quaisquer dependências específicas do Panorama APIs, relacionadas principalmente à conexão e streaming da câmera. AWS forneceu um exemplo de código para mostrar como fazer essas alterações. Depois que essas dependências forem removidas, o aplicativo poderá ser movido para uma plataforma de hardware alternativa.

Se eu tiver problemas em ou após 20 de maio de 2025, qual suporte estará disponível?

AWS continuará a fornecer suporte para o Panorama até o final do período de notificação de descontinuação (31 de maio de 2026). Para quaisquer requisitos de suporte, os clientes devem inserir um caso de suporte por meio de seus canais de suporte normais. AWS fornecerá atualizações de segurança, correções de erros e aprimoramentos de disponibilidade.

Não posso migrar antes de 31 de maio de 2026. A data pode ser estendida?

Estamos confiantes de que as alternativas disponíveis para o Panorama permitem que os clientes migrem para uma solução alternativa até 31 de maio de 2026 e não temos planos de estender a disponibilidade do serviço após essa data.

Meu aplicativo Edge continuará funcionando após o término do serviço?

Não. O dispositivo e os aplicativos Panorama dependem da conectividade com o serviço de nuvem Panorama. Depois que esse serviço for descontinuado em 31 de maio de 2026, nem o aplicativo Panorama nem o dispositivo Panorama continuarão funcionando.

Começando com AWS Panorama

Para começar AWS Panorama, primeiro conheça os conceitos do serviço e a terminologia usada neste guia. Em seguida, você pode usar o AWS Panorama console para registrar seu AWS Panorama equipamento e criar um aplicativo. Em cerca de uma hora, você pode configurar o dispositivo, atualizar o software e implantar uma aplicação de exemplo. Para concluir os tutoriais desta seção, você usa o AWS Panorama equipamento e uma câmera que transmite vídeo por uma rede local.

> **Note**
>
> Para comprar um AWS Panorama aparelho, acesse o AWS Panorama console.

O aplicativo AWS Panorama de amostra demonstra o uso de AWS Panorama recursos. Ele inclui um modelo que foi treinado com SageMaker IA e um código de amostra que usa o SDK do AWS Panorama aplicativo para executar inferência e gerar vídeo. O aplicativo de amostra inclui um AWS CloudFormation modelo e scripts que mostram como automatizar fluxos de trabalho de desenvolvimento e implantação a partir da linha de comando.

Os dois tópicos finais deste capítulo detalham os requisitos para modelos e câmeras e as especificações de hardware do AWS Panorama Appliance. Se você ainda não adquiriu um dispositivo e câmeras, ou planeja desenvolver seus próprios modelos de visão computacional, consulte esses tópicos antes, para obter mais informações.

Tópicos

Conceitos do AWS Panorama

No AWS Panorama, você cria aplicações de visão computacional e as implanta no AWS Panorama Appliance ou em um dispositivo compatível para analisar streams de vídeo de câmeras de rede. Você escreve o código da aplicação em Python e cria contêineres de aplicações com o Docker. Você usa a CLI do AWS Panorama Application para importar modelos de machine learning localmente ou do Amazon Simple Storage Service (Amazon S3). As aplicações usam o SDK para aplicações do AWS Panorama para receber entrada de vídeo de uma câmera e interagir com um modelo.

Conceitos

O AWS Panorama Appliance

O AWS Panorama Appliance é o hardware que executa suas aplicações. Você usa o console do AWS Panorama para registrar um dispositivo, atualizar seu software e implantar aplicações nele. O software no AWS Panorama Appliance se conecta aos streams de câmeras, envia quadros de vídeo para sua aplicação e exibe a saída de vídeo em um monitor anexado.

O AWS Panorama Appliance é um dispositivo de borda desenvolvido pela Nvidia Jetson AGX Xavier. Em vez de enviar imagens para a AWS nuvem para processamento, ele executa aplicativos localmente em hardware otimizado. Isso permite que você analise o vídeo em tempo real e processe os resultados localmente. O dispositivo requer uma conexão com a Internet para relatar seu status, fazer upload de logs e realizar atualizações e implantações de software.

Para obter mais informações, consulte Gerenciando o AWS Panorama equipamento.

Dispositivos compatíveis

Além do AWS Panorama Appliance, o AWS Panorama oferece suporte a dispositivos compatíveis de AWS parceiros. Os dispositivos compatíveis oferecem suporte aos mesmos atributos do AWS Panorama Appliance. Você pode registrar e gerenciar dispositivos compatíveis e criar e implantar aplicações usando o console e a API do AWS Panorama.

- Lenovo ThinkEdge® SE7 0 — Desenvolvido pela Nvidia Jetson Xavier NX

O conteúdo e as aplicações de exemplo deste guia são desenvolvidos com o AWS Panorama Appliance. Para obter mais informações sobre atributos específicos de hardware e software para seu dispositivo, consulte a documentação do fabricante.

Aplicações

As aplicações são executadas no AWS Panorama Appliance para realizar tarefas de visão computacional em streams de vídeo. Você pode criar aplicações de visão computacional combinando código Python e modelos de machine learning e implantá-los no AWS Panorama Appliance pela Internet. As aplicações podem enviar vídeo para um monitor ou usar o SDK da AWS para enviar resultados para os serviços da AWS.

Para criar e implantar aplicações, use a CLI da aplicação do AWS Panorama. A CLI da aplicação do AWS Panorama é uma ferramenta de linha de comando que gera pastas de aplicações e arquivos de configuração padrão, cria contêineres com o Docker e faz uploads de ativos. Você pode executar várias aplicações em um dispositivo.

Para obter mais informações, consulte Gerenciando AWS Panorama aplicativos.

Nós

Uma aplicação compreende vários componentes chamados nós, que representam entradas, saídas, modelos e código. Um nó pode ser apenas de configuração (entradas e saídas) ou incluir artefatos (modelos e código). Os nós de código de uma aplicação são agrupados em pacotes de nós e carregados para um ponto de acesso Amazon S3, onde o AWS Panorama Appliance pode acessá-los. Um manifesto da aplicação é um arquivo de configuração que define as conexões entre os nós.

Para obter mais informações, consulte Nós da aplicação.

Modelos da

Um modelo de visão computacional é uma rede de machine learning treinada para processar imagens. Os modelos de visão computacional podem realizar várias tarefas, como classificação, detecção, segmentação e rastreamento. Um modelo de visão computacional usa uma imagem como entrada e gera informações sobre a imagem ou os objetos na imagem.

O AWS Panorama oferece suporte a modelos criados com PyTorch MXNet, Apache e. TensorFlow Você pode criar modelos com a Amazon SageMaker AI ou em seu ambiente de desenvolvimento. Para obter mais informações, consulte ???.

Configuração do AWS Panorama Appliance

Para começar a usar seu AWS Panorama Appliance ou dispositivo compatível, registre-o no console do AWS Panorama e atualize seu software. Durante o processo de configuração, você cria um recurso de dispositivo no AWS Panorama que representa o dispositivo físico e copia arquivos para o dispositivo usando uma unidade USB. O dispositivo usa esses certificados e arquivos de configuração para se conectar ao serviço AWS Panorama. Em seguida, você usa o console do AWS Panorama para atualizar o software do dispositivo e registrar as câmeras.

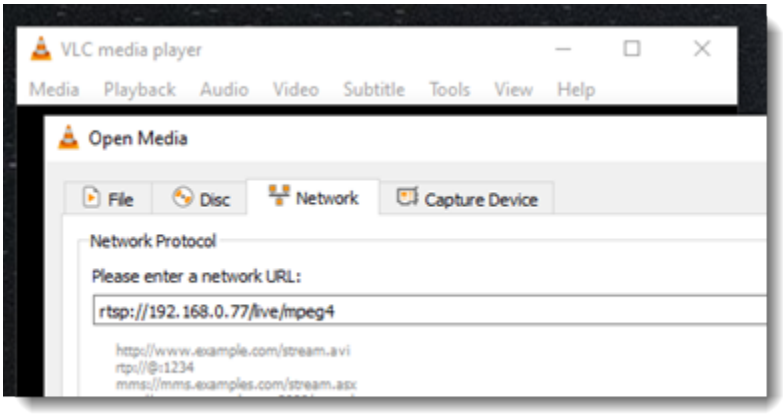
Seções

Pré-requisitos

Para seguir este tutorial, você precisa de um AWS Panorama Appliance ou de um dispositivo compatível e do seguinte hardware:

- Monitor: um monitor com entrada HDMI para visualizar a saída da aplicação de exemplo.
- Unidade USB (incluída no AWS Panorama Appliance) — Uma unidade FAT32 de memória flash USB 3.0 formatada com pelo menos 1 GB de armazenamento, para transferir um arquivo com arquivos de configuração e um certificado para o AWS Panorama Appliance.
- Câmera: uma câmera IP que envia um stream de vídeo RTSP.

Use as ferramentas e instruções fornecidas pelo fabricante da câmera para identificar o endereço IP e o caminho de transmissão da câmera. Você pode usar um reprodutor de vídeo, como o VLC, para verificar o URL do stream, abrindo-o como uma fonte de mídia de rede:



O console do AWS Panorama usa outros serviços da AWS para montar componentes de aplicações, gerenciar permissões e verificar configurações. Para registrar um dispositivo e implantar a aplicação de exemplo, você precisa das seguintes permissões:

- AWSPanoramaFullAccess— Fornece acesso total ao AWS Panorama, aos pontos de acesso do AWS Panorama no Amazon S3, às credenciais do dispositivo e aos registros do dispositivo na AWS Secrets Manager Amazon. CloudWatch Inclui permissão para criar um perfil vinculado ao serviço para o AWS Panorama.
- AWS Identity and Access Management (IAM) — Na primeira execução, para criar funções usadas pelo serviço AWS Panorama e pelo AWS Panorama Appliance.

Se você não tiver permissão para criar perfis no IAM, peça a um administrador que abra o console do AWS Panorama e aceite a solicitação para criar perfis de serviço.

Registro e configuração do AWS Panorama Appliance

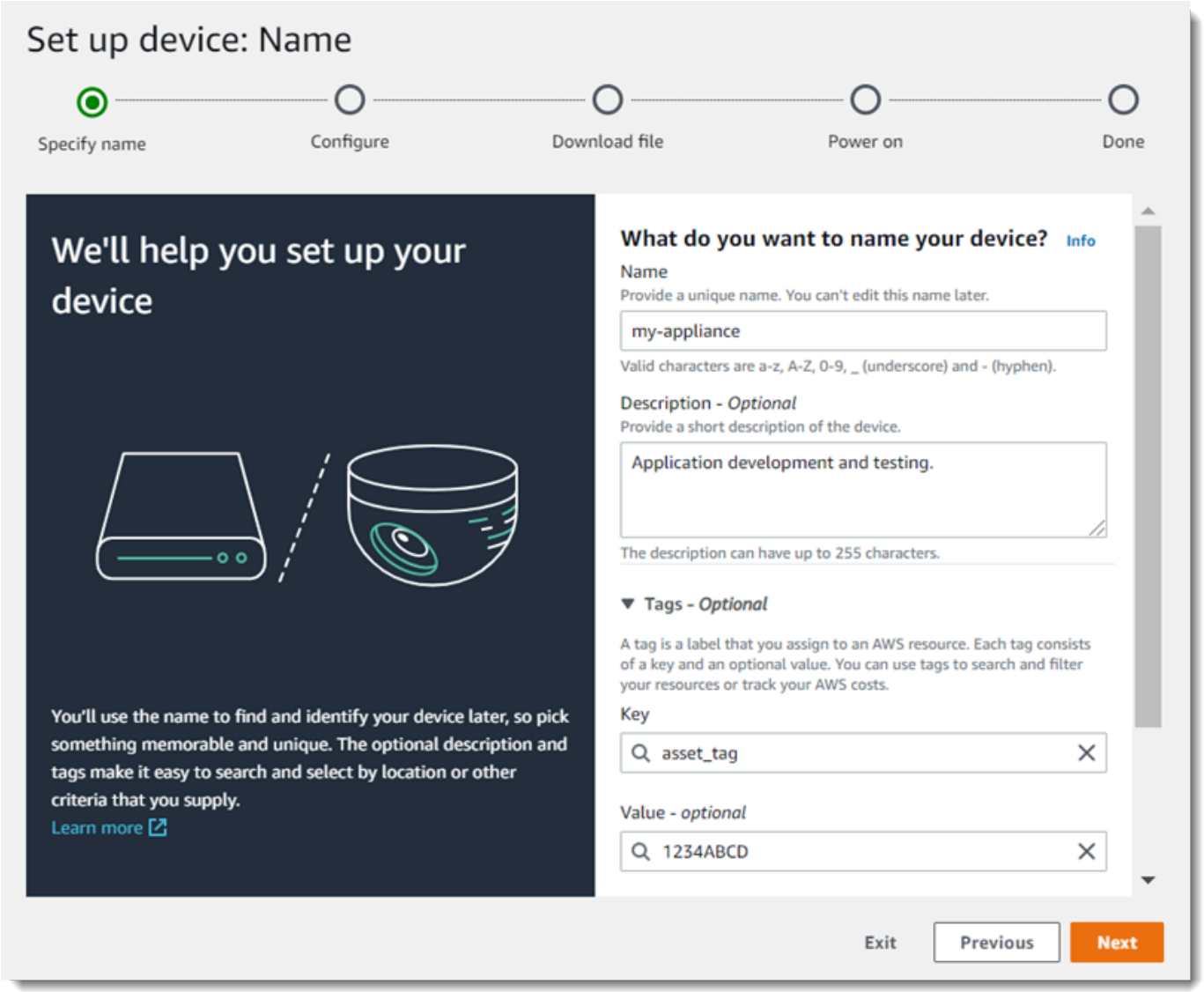
O AWS Panorama Appliance é um dispositivo de hardware que se conecta a câmeras habilitadas para rede por meio de uma conexão de rede local. Ele usa um sistema operacional baseado em Linux que inclui o SDK para aplicações do AWS Panorama e software de suporte para execução de aplicações de visão computacional.

Para se conectar ao AWS gerenciamento do equipamento e à implantação do aplicativo, o equipamento usa um certificado de dispositivo. Você usa o console do AWS Panorama para gerar um certificado de provisionamento. O dispositivo usa esse certificado temporário para concluir a configuração inicial e baixar um certificado de dispositivo permanente.

> **Important**
>
> O certificado de provisionamento que você gera nesse procedimento só é válido por 5 minutos. Se você não concluir o processo de registro dentro desse prazo, deverá começar de novo.

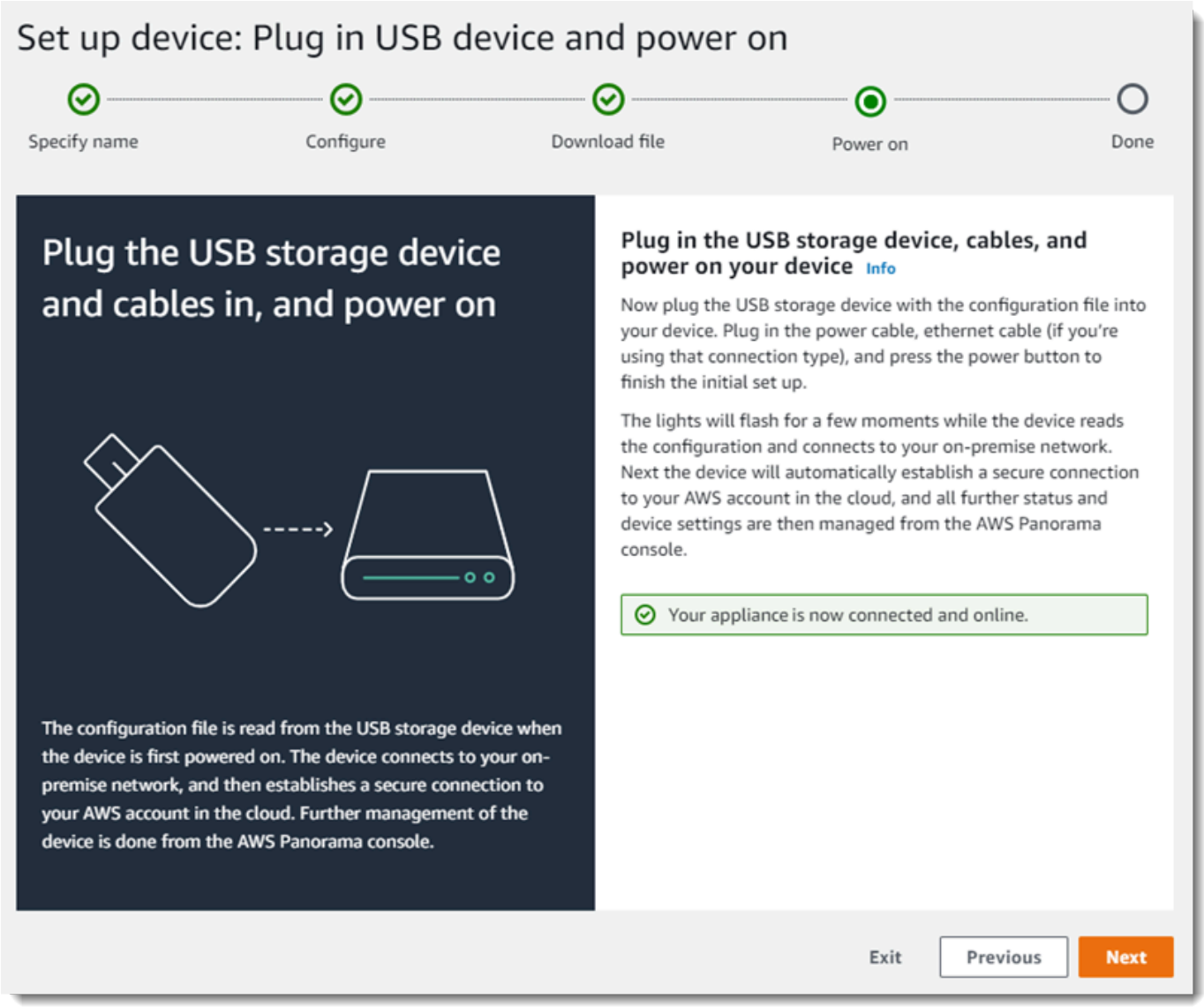
Para registrar um dispositivo

1. Conecte a unidade USB ao seu computador. Prepare o dispositivo conectando os cabos de rede e de alimentação. O dispositivo é ligado e aguarda a conexão de uma unidade USB.
2. Abra a Página de conceitos básicos do console do AWS Panorama.
3. Escolha Adicionar dispositivo.
4. Escolha Iniciar configuração.
5. Insira um nome e uma descrição para o recurso de dispositivo que representa o dispositivo no AWS Panorama. Escolha Próximo.



6. Se você precisar atribuir manualmente um endereço IP, um servidor NTP ou configurações de DNS, escolha Configurações de rede avançadas. Caso contrário, escolha Next.
7. Escolha Baixar arquivo. Escolha Próximo.
8. Copie o arquivo de configuração para o diretório raiz da unidade USB.
9. Conecte a unidade USB à porta USB 3.0 na parte frontal do aparelho, ao lado da porta HDMI.

   Quando você conecta a unidade USB, o equipamento copia o arquivo de configuração e o arquivo de configuração de rede para si mesmo e se conecta à AWS nuvem. A luz de status do dispositivo muda de verde para azul quando ele conclui a conexão, e depois volta para verde.
10. Para continuar, escolha Avançar.



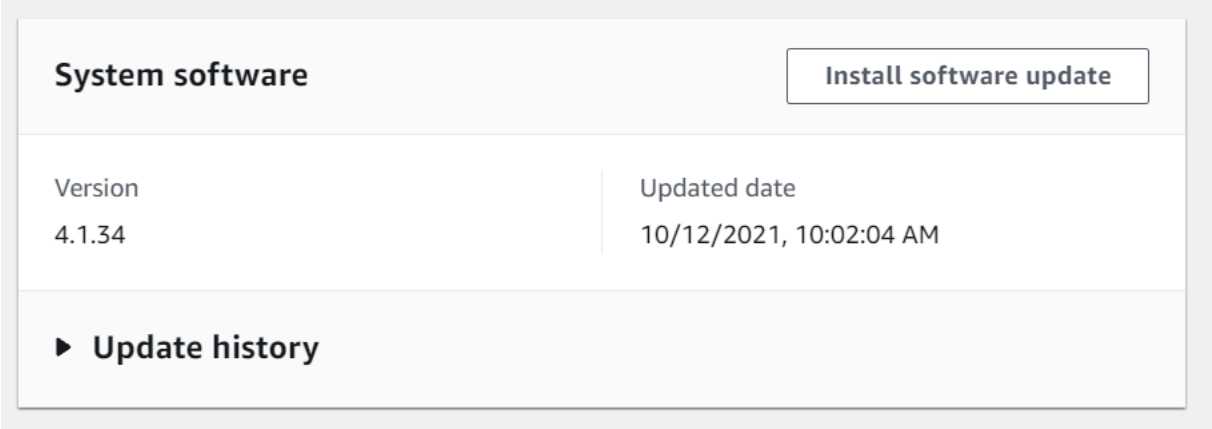
11. Selecione Concluído.

Atualize o software do dispositivo

O AWS Panorama Appliance tem vários componentes de software, incluindo um sistema operacional Linux, o SDK para aplicações do AWS Panorama e o suporte de bibliotecas e estruturas de visão computacional. Para garantir que você possa usar os atributos e aplicações mais recentes com seu dispositivo, atualize o software após a configuração e sempre que houver uma atualização disponível.

Para atualizar o software do dispositivo

1. Abra a página Dispositivos do console do AWS Panorama.
2. Escolha um dispositivo.
3. Escolha Configurações
4. Em Software do sistema, escolha Instalar atualização de software.



5. Escolha uma nova versão e, em seguida, selecione Instalar.

> **Important**
>
> Antes de continuar, remova a unidade USB do dispositivo e formate-a para excluir seu conteúdo. O arquivo de configuração contém dados confidenciais e não é excluído automaticamente.

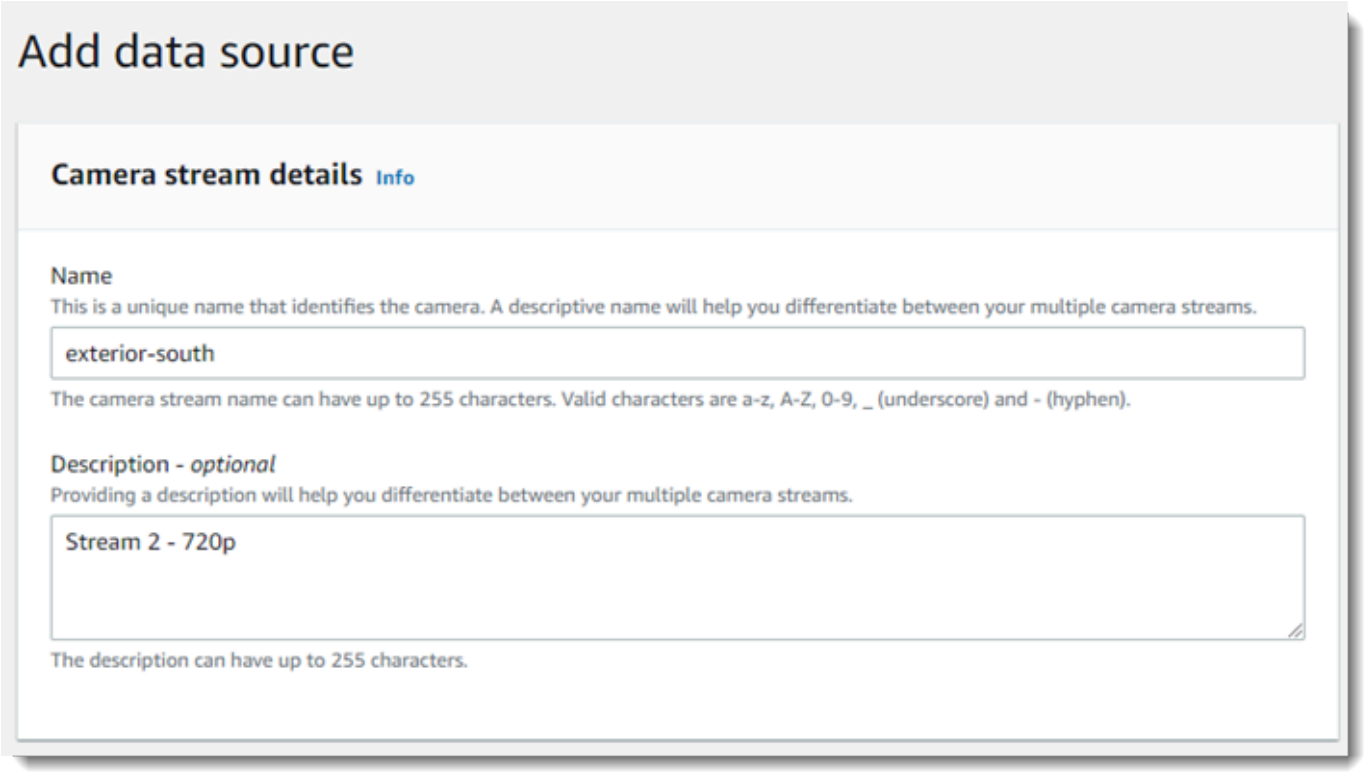
O processo de atualização pode demorar 30 minutos ou mais para ser concluído. Você pode monitorar seu progresso no console do AWS Panorama ou em um monitor conectado. Quando o processo for concluído, o dispositivo será reinicializado.

Adição de um stream de câmera

Em seguida, registre um stream de câmera no console do AWS Panorama.

Para registrar um stream de câmera

1. Abra a Página de fontes de dados do console do AWS Panorama.
2. Escolha Adicionar fonte de dados.



3. Configure as definições a seguir.

   - Nome: um nome para o stream da câmera.
   - Descrição: uma breve descrição da câmera, sua localização ou outros detalhes.
   - URL RTSP: URL que especifica o endereço IP da câmera e o caminho para o stream. Por exemplo, `rtsp://192.168.0.77/live/mpeg4/`
   - Credenciais: se o stream da câmera estiver protegido por senha, especifique o nome de usuário e a senha.
4. Escolha Salvar.

O AWS Panorama armazena as credenciais da sua câmera com segurança em. AWS Secrets Manager Várias aplicações podem processar o mesmo stream de câmera simultaneamente.

Próximas etapas

Se você encontrou erros durante a configuração, consulte Solução de problemas.

Para implantar uma aplicação de exemplo, vá para o próximo tópico.

Implantação da aplicação de exemplo do AWS Panorama

Depois de configurar seu AWS Panorama Appliance ou dispositivo compatível e atualizar seu software, implante uma aplicação de exemplo. As seções a seguir mostram como importar uma aplicação de exemplo com a CLI da aplicação do AWS Panorama e implantá-la com o console do AWS Panorama.

A aplicação de exemplo usa um modelo de machine learning para classificar objetos em quadros de vídeo de uma câmera de rede. Ela usa o SDK para aplicações do AWS Panorama para carregar um modelo, obter imagens e executar o modelo. Em seguida, a aplicação sobrepõe os resultados ao vídeo original e os envia para um monitor conectado.

Em um ambiente de varejo, a análise dos padrões de tráfego de pedestres permite prever os níveis de tráfego. Ao combinar a análise com outros dados, você pode se preparar para a maior necessidade de pessoal em feriados e outros eventos, medir a eficácia de anúncios e promoções de vendas ou otimizar o posicionamento do monitor e o gerenciamento de inventário.

Seções

Pré-requisitos

Para seguir os procedimentos deste tutorial, você precisa de um terminal de linha de comando ou de um shell para executar comandos. Nas listagens de código, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

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

Para comandos longos, um caractere de escape (\) é usado para dividir um comando em várias linhas.

No Linux e no macOS, use seu gerenciador preferido de pacotes e de shell. No Windows 10, você pode instalar o Subsistema Windows para Linux para obter uma versão do Ubuntu integrada com o Windows e o Bash. Para obter ajuda na configuração de um ambiente de desenvolvimento no Windows, consulte Configurar um ambiente de desenvolvimento no Windows.

Você usa o Python para desenvolver aplicações do AWS Panorama e instalar ferramentas com o pip, o gerenciador de pacotes do Python. Se você ainda não tiver Python, instale a versão mais recente. Se você tiver o Python 3, mas não o pip, instale o pip com o gerenciador de pacotes do seu sistema operacional ou instale uma nova versão do Python, que vem com o pip.

Neste tutorial, você usa o Docker para criar o contêiner que executa o código da sua aplicação. Instale o Docker, a partir do website do Docker: Obter Docker.

Este tutorial usa a CLI da aplicação do AWS Panorama para importar a aplicação de exemplo, criar pacotes e fazer upload de artefatos. A CLI do aplicativo AWS Panorama usa o AWS Command Line Interface (AWS CLI) para chamar as operações de API do serviço. Se você já tiver o AWS CLI, atualize-o para a versão mais recente. Para instalar a CLI do aplicativo AWS Panorama e AWS CLI, use. `pip`

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

Baixe a aplicação de exemplo e extraia-a em seu espaço de trabalho.

- Exemplo de aplicativo — aws-panorama-sample.zip

Importe a aplicação de exemplo

Para importar a aplicação de exemplo para uso em sua conta, use a CLI da aplicação do AWS Panorama. As pastas e o manifesto da aplicação contêm referências a um número de conta reservado. Para atualizá-los com o número da sua conta, execute o comando `panorama-cli import-application`.

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

O pacote `SAMPLE_CODE`, no diretório `packages`, contém o código e a configuração da aplicação, incluindo um Dockerfile que usa a imagem base da aplicação, `panorama-application`. Para criar o contêiner da aplicação que é executado no dispositivo, use o 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
```

A etapa final com a CLI da aplicação do AWS Panorama é registrar o código e os nós do modelo da aplicação e fazer o upload dos ativos em um ponto de acesso Amazon S3 fornecido pelo serviço. Os ativos incluem a imagem do contêiner do código, o modelo e um arquivo descritor para cada um. Para registrar os nós e fazer o upload dos ativos, execute o 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
```

Implantar a aplicação

Use o console do AWS Panorama para implantar a aplicação em seu dispositivo.

Para implantar a aplicação

1. Abra a Página de aplicações implantadas do console do AWS Panorama.
2. Escolha Implantar aplicação.
3. Cole o conteúdo do manifesto da aplicação, `graphs/aws-panorama-sample/graph.json`, no editor de texto. Escolha Próximo.
4. Em Application name (Nome da aplicação), insira `aws-panorama-sample`.
5. Escolha Prosseguir para a implantação.
6. Escolha Iniciar implantação.
7. Escolha Avançar sem selecionar um perfil.
8. Escolha Selecionar dispositivo e, em seguida, escolha seu dispositivo. Escolha Próximo.
9. Na etapa Selecionar fontes de dados, escolha Visualizar entrada(s) e adicione o stream da câmera como uma fonte de dados. Escolha Próximo.

10. Na etapa Configurar, escolha Avançar.
11. Escolha Implantar e Concluído.
12. Na lista de aplicativos implantados, escolha aws-panorama-sample.

Atualize esta página para obter atualizações ou use o script a seguir para monitorar a implantação na linha de comando.

Example monitor-deployment.sh

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

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

```
    }
]
...
```

Se o aplicativo não começar a ser executado, verifique os registros do aplicativo e do dispositivo no Amazon CloudWatch Logs.

Visualizar a saída

Quando a implantação estiver concluída, o aplicativo começará a processar o stream de vídeo e enviará os registros para CloudWatch o.

Para ver registros em CloudWatch Registros

1. Abra a página Grupos de CloudWatch registros do console de registros.
2. Encontre logs de aplicações e dispositivos do AWS Panorama nos seguintes grupos:

   - Logs do dispositivo: `/aws/panorama/devices/device-id`
   - Logs de aplicações: `/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 visualizar a saída de vídeo da aplicação, conecte o dispositivo a um monitor com um cabo HDMI. Por padrão, a aplicação mostra qualquer resultado de classificação com mais de 20% de confiança.

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",
...
```

O modelo de exemplo tem 1.000 classes, incluindo muitos animais, alimentos e objetos comuns. Tente apontar sua câmera para um teclado ou uma caneca de café.

Para simplificar, a aplicação de exemplo usa um modelo de classificação leve. O modelo gera uma única matriz com uma probabilidade para cada uma de suas classes. As aplicações do mundo real usam com mais frequência modelos de detecção de objetos que têm saída multidimensional. Para exemplos de aplicações com modelos mais complexos, consulte Exemplos de aplicações, scripts e modelos.

Habilitar o SDK para Python

O aplicativo de amostra usa o AWS SDK for Python (Boto) para enviar métricas para a Amazon CloudWatch. Para habilitar essa funcionalidade, crie um perfil que conceda permissão à aplicação para enviar métricas e reimplante a aplicação com o perfil anexado.

O aplicativo de amostra inclui um AWS CloudFormation modelo que cria uma função com as permissões necessárias. Para criar uma função, use o comando `aws cloudformation deploy`.

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

Para reimplantar a aplicação

1. Abra a Página de aplicações implantadas do console do AWS Panorama.
2. Escolha a aplicação.
3. Selecione Replace (Substituir).
4. Conclua as etapas para implantar a aplicação. Em Especificar perfil do IAM, escolha a função que você criou. O nome começa com `aws-panorama-sample-runtime`.
5. Quando a implantação for concluída, abra o CloudWatchconsole e visualize as métricas no `AWSPanoramaApplication` namespace. A cada 150 quadros, a aplicação registra e carrega métricas para processamento de quadros e tempo de inferência.

Limpeza

Quando você terminar de trabalhar com a aplicação de exemplo, poderá usar o console do AWS Panorama para removê-la do dispositivo.

Para remover uma aplicação do dispositivo

1. Abra a Página de aplicações implantadas do console do AWS Panorama.

2. Escolha a aplicação.
3. Escolha Excluir do dispositivo.

Próximas etapas

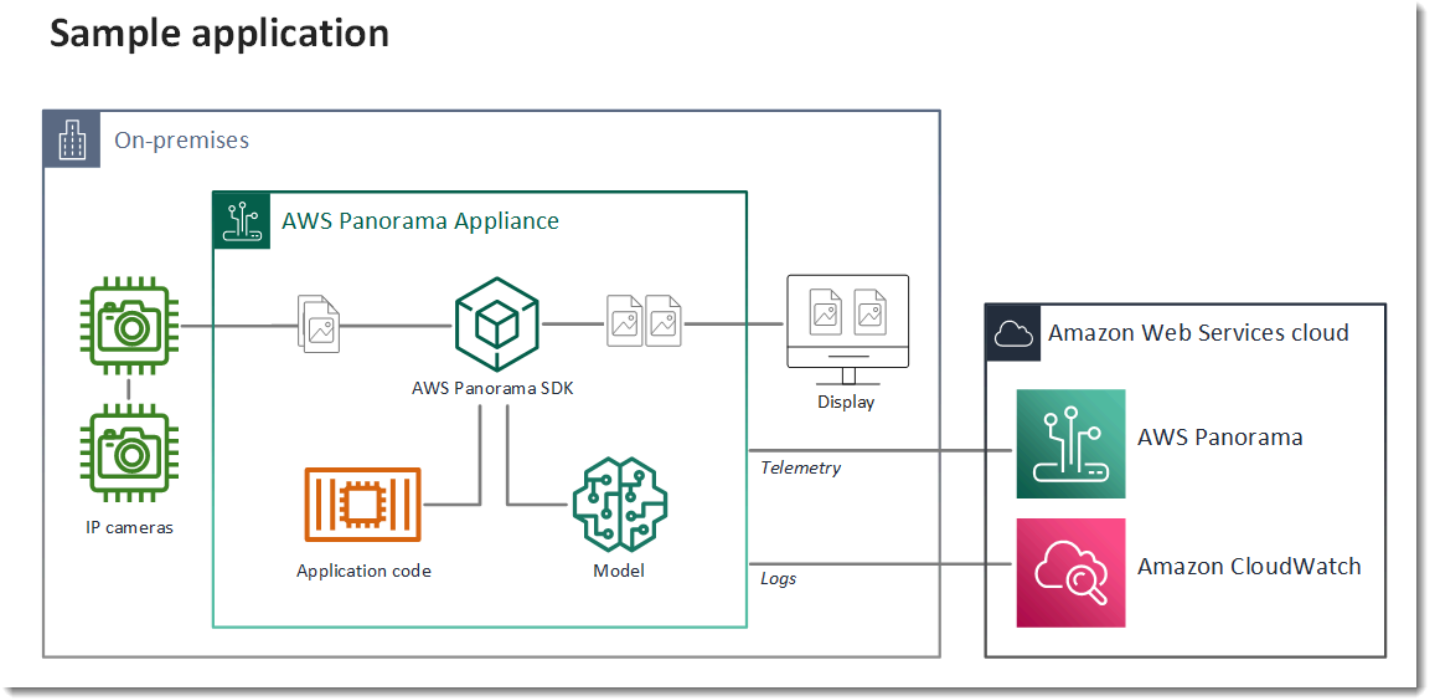
Se você encontrar erros ao implantar ou executar a aplicação de exemplo, consulte Solução de problemas.

Para saber mais sobre os atributos e a implementação da aplicação de exemplo, continue com o próximo tópico.

Desenvolvimento de aplicativos do AWS Panorama

Você pode usar a aplicação de exemplo para saber como usar a estrutura da aplicação do AWS Panorama e como ponto de partida para sua própria aplicação.

O diagrama a seguir mostra os principais componentes da aplicação em execução em um dispositivo AWS Panorama. O código da aplicação usa o SDK para aplicações do AWS Panorama para obter imagens e interagir com o modelo, ao qual ele não tem acesso direto. A aplicação envia vídeos para um monitor conectado, mas não envia dados de imagem para fora da rede local.



Neste exemplo, a aplicação usa o SDK para aplicações do AWS Panorama para obter quadros de vídeo de uma câmera, pré-processar os dados do vídeo e enviar os dados para um modelo de visão computacional que detecta objetos. A aplicação exibe o resultado em um monitor HDMI conectado ao aparelho.

Seções

- Próximas etapas

O manifesto da aplicação

O manifesto da aplicação é um arquivo chamado `graph.json` na `graphs` pasta. O manifesto define os componentes da aplicação, que são pacotes, nós e bordas.

Os pacotes são arquivos binários, de código e de configuração com informações de código de aplicação, modelos, câmeras e monitores. O aplicativo de exemplo usa 4 pacotes:

Example Pacotes do **`graphs/aws-panorama-sample/graph.json`**

```
    "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"
        }
    ],
```

Os dois primeiros pacotes são definidos na aplicação, no diretório `packages`. Eles contêm o código e o modelo específicos dessa aplicação. Os dois segundos pacotes são pacotes genéricos de câmeras e monitor fornecidos pelo serviço AWS Panorama. O pacote `abstract_rtsp_media_source` é um espaço reservado para uma câmera, que é substituído durante a implantação. O pacote `hdmi_data_sink` representa o conector de saída HDMI no dispositivo.

Os nós são interfaces para pacotes, bem como parâmetros que não são de pacotes e que podem ter valores padrão que serão substituídos no momento da implantação. Os pacotes de código e modelo

definem interfaces em arquivos `package.json` que especificam entradas e saídas, que podem ser streams de vídeo ou um tipo de dados básico, como float, booleano ou string.

Por exemplo, o nó `code_node` se refere a uma interface do pacote `SAMPLE_CODE`.

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

Essa interface é definida no arquivo de configuração do pacote, `package.json`. A interface especifica que o pacote é uma lógica de negócios e que usa um stream de vídeo chamado `video_in` e um número de ponto flutuante chamado `threshold` como entradas. A interface também especifica que o código requer um buffer de stream de vídeo chamado `video_out` para enviar vídeo para um monitor.

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

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

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

De volta ao manifesto da aplicação, o nó `camera_node` representa um stream de vídeo de uma câmera. Ele inclui um decorador que aparece no console quando você implanta a aplicação, solicitando que você escolha um stream de uma câmera.

Example Nós de câmera do **`graphs/aws-panorama-sample/graph.json`**

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

Um nó de parâmetro `threshold_param` define o parâmetro de limite de confiança usado pelo código da aplicação. Ele tem um valor padrão de 60 e pode ser substituído durante a implantação.

Example Nó de parâmetro do **`graphs/aws-panorama-sample/graph.json`**

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

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

A seção final do manifesto da aplicação, `edges`, faz conexões entre os nós. O stream de vídeo da câmera e o parâmetro de limite se conectam à entrada do nó do código, e a saída de vídeo do nó do código se conecta ao monitor.

Example Bordas do **`graphs/aws-panorama-sample/graph.json`**

```
        "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"
            }
        ]
```

Compilação com a aplicação de exemplo

Você pode usar a aplicação exemplo como um ponto de partida para criar sua própria aplicação.

O nome de cada pacote deve ser exclusivo em sua conta. Se você e outro usuário da sua conta usarem um nome de pacote genérico, como `code` ou`model`, você poderá obter a versão errada do pacote ao implantar. Altere o nome do pacote de código para um nome que represente sua aplicação.

Para renomear o pacote de código

1. Renomeie a pasta do pacote: `packages/123456789012-SAMPLE_CODE-1.0/`.
2. Atualize o nome do pacote nos seguintes locais.

   - Manifesto da aplicação: `graphs/aws-panorama-sample/graph.json`

- Configuração do pacote: `packages/123456789012-SAMPLE_CODE-1.0/package.json`
- Script de compilação: `3-build-container.sh`

Para atualizar o código do aplicativo

1. Modifique o código da aplicação em `packages/123456789012-SAMPLE_CODE-1.0/src/application.py`.
2. Para criar o contêiner, execute `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
```

A CLI exclui automaticamente o ativo de contêiner antigo da pasta `assets` e atualiza a configuração do pacote.

3. Para carregar os pacotes, execute `4-package-application.py`.

4. Abra a Página de aplicações implantadas do console do AWS Panorama.
5. Escolha a aplicação.
6. Selecione Replace (Substituir).
7. Conclua as etapas para implantar a aplicação. Se necessário, você pode fazer alterações no manifesto da aplicação, nos streams da câmera ou nos parâmetros.

Alteração do modelo de visão computacional

A aplicação de exemplo inclui um modelo de visão computacional. Para usar seu próprio modelo, modifique a configuração do nó do modelo e use a CLI da aplicação do AWS Panorama para importá-lo como um ativo.

O exemplo a seguir usa um modelo MXNet SSD ResNet 5.0 que você pode baixar do GitHub repositório deste guia: ssd_512_resnet50_v1_voc.tar.gz

Para alterar o modelo da aplicação de exemplo

1. Renomeie a pasta do pacote de forma que o nome dela corresponda ao seu modelo. Por exemplo, para `packages/123456789012-SSD_512_RESNET50_V1_VOC-1.0/`.
2. Atualize o nome do pacote nos seguintes locais.

   - Manifesto da aplicação: `graphs/aws-panorama-sample/graph.json`
   - Configuração do pacote: `packages/123456789012-SSD_512_RESNET50_V1_VOC-1.0/package.json`
3. No arquivo de configuração do pacote (`package.json`). Altere o valor `assets` para uma matriz em branco.

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

4. Abra o arquivo descritor do pacote (`descriptor.json`). Atualize os valores `framework` e `shape` para que correspondam ao seu modelo.

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

O valor da forma, `1,3,512,512`, indica o número de imagens que o modelo usa como entrada (1), o número de canais em cada imagem (3- vermelho, verde e azul) e as dimensões da imagem (512 x 512). Os valores e a ordem da matriz variam entre os modelos.

5. Importe o modelo com a CLI da aplicação do AWS Panorama. A CLI da aplicação do AWS Panorama copia os arquivos de modelo e descritor na pasta `assets` com nomes exclusivos e atualiza a configuração do pacote.

```
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 fazer o upload do modelo, execute `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. Atualize o código do aplicativo. A maior parte do código pode ser reutilizada. O código específico para a resposta do modelo está no 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)
```

Dependendo do seu modelo, talvez você também precise atualizar o método `preprocess`.

Pré-processamento de imagens

Antes de a aplicação enviar uma imagem para o modelo, ela a prepara para inferência redimensionando-a e normalizando os dados de cores. O modelo usado pela aplicação requer uma imagem de 224 x 224 pixels com três canais de cores, para corresponder ao número de entradas em sua primeira camada. A aplicação ajusta cada valor de cor convertendo-o em um número entre 0 e 1, subtraindo o valor médio dessa cor e dividindo o resultado pelo desvio padrão. Por fim, ele combina os canais de cores e os converte em uma NumPy matriz que o modelo pode processar.

Example application.py: pré-processamento

```
    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)
```

Esse processo fornece os valores do modelo em uma faixa previsível centrada em torno de 0. O processo corresponde ao pré-processamento aplicado às imagens no conjunto de dados de treinamento, que é uma abordagem padrão, mas pode variar de acordo com o modelo.

Upload de métricas com o SDK para Python

O aplicativo de amostra usa o SDK para Python para fazer upload de métricas para a 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.")
```

Ela obtém permissão de uma função de runtime que você atribui durante a implantação. A função é definida no `aws-panorama-sample.yml` AWS CloudFormation modelo.

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/
```

A aplicação de exemplo instala o SDK para Python e outras dependências com pip. Quando você cria o contêiner da aplicação, o `Dockerfile` executa comandos para instalar bibliotecas acima do que vem com a imagem 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 o AWS SDK no código do seu aplicativo, primeiro modifique o modelo para adicionar permissões para todas as ações de API que o aplicativo usa. Atualize a AWS CloudFormation pilha executando o `1-create-role.sh` sempre que fizer uma alteração. Em seguida, implante as alterações no código da sua aplicação.

Para ações que modificam ou usam recursos existentes, é uma prática recomendada minimizar o escopo dessa política especificando um nome ou padrão para o destino `Resource` em uma declaração separada. Para obter detalhes sobre as ações e os recursos compatíveis com cada serviço, consulte Ação, recursos e chaves de condição na Referência de autorização do serviço

Próximas etapas

Para obter instruções sobre como usar a CLI da aplicação do AWS Panorama para criar aplicações e pacotes do zero, consulte o README da CLI.

- github. com/aws/aws-clipe panorâmico

Para obter mais exemplos de código e um utilitário de teste que você pode usar para validar o código da sua aplicação antes da implantação, visite o repositório de exemplos do AWS Panorama.

- github. com/aws-samples/aws-amostras panorâmicas

Modelos e câmeras de visão computacional compatíveis

O AWS Panorama oferece suporte a modelos criados com PyTorch MXNet, Apache e. TensorFlow Quando você implanta um aplicativo, o AWS Panorama compila seu modelo no SageMaker AI Neo. Você pode criar modelos na Amazon SageMaker AI ou em seu ambiente de desenvolvimento, desde que use camadas compatíveis com o SageMaker AI Neo.

Para processar vídeos e enviar imagens para um modelo, o AWS Panorama Appliance se conecta a um stream de vídeo codificado em H.264 com o protocolo RTSP. O AWS Panorama testa diversas câmeras comuns para verificar a compatibilidade.

Seções

Modelos compatíveis

Ao criar uma aplicação para o AWS Panorama, você fornece um modelo de machine learning que a aplicação usa para visão computacional. Você pode usar modelos pré-criados e pré-treinados fornecidos por estruturas de modelo, um modelo de exemplo ou um modelo que você mesmo cria e treina.

> **Note**
>
> Para obter uma lista de modelos pré-criados que foram testados com o AWS Panorama, consulte Compatibilidade de modelos.

Quando você implanta um aplicativo, o AWS Panorama usa o compilador SageMaker AI Neo para compilar seu modelo de visão computacional. SageMaker O AI Neo é um compilador que otimiza modelos para serem executados com eficiência em uma plataforma de destino, que pode ser uma instância no Amazon Elastic Compute Cloud (Amazon EC2) ou um dispositivo de ponta, como o AWS Panorama Appliance.

O AWS Panorama oferece suporte às versões do PyTorch Apache MXNet e TensorFlow que são compatíveis com dispositivos de ponta pelo SageMaker AI Neo. Ao criar seu próprio modelo, você pode usar as versões da estrutura listadas nas notas de lançamento do SageMaker AI Neo. Na SageMaker IA, você pode usar o algoritmo de classificação de imagens incorporado.

Para obter mais informações sobre o uso de modelos no AWS Panorama, consulte Modelos de visão computacional.

Câmeras compatíveis

O AWS Panorama Appliance suporta streams de vídeo H.264 de câmeras com saída RTSP em uma rede local. Para streams de câmera maiores que 2 megapixels, o dispositivo reduz a imagem para 1920 x 1080 pixels ou um tamanho equivalente que preserva a proporção do stream.

Os seguintes modelos de câmera foram testados quanto à compatibilidade com o AWS Panorama Appliance:

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

Para obter as especificações de hardware do dispositivo, consulte Especificações do AWS Panorama Appliance.

Especificações do AWS Panorama Appliance

O AWS Panorama Appliance tem as seguintes especificações de hardware. Para conhecer outros dispositivos compatíveis, consulte a documentação do fabricante.

Componente	Especificação
Processador e GPU	Nvidia Jetson AGX Xavier com 32 GB de RAM
Ethernet	2x 1000 Base-T (Gigabyte)
USB	1x USB 2.0 e 1x USB 3.0 tipo A fêmea
Saída do HDMI	2.0a
Dimensões	197 mm x 243 mm x 40 mm
Weight	1,7 kg
Fonte de alimentação	100 a 240 V 50 a 60 Hz CA 65 W
Entrada de energia	Tomada IEC 60320 C6 (3 pinos)
Proteção contra poeira e líquidos	IP-62
Conformidade regulatória EMI/EMC	FCC Parte 15 (EUA)
Limites do toque térmico	IEC-62368
Temperatura de operação	-20°C a 60°C
Umidade operacional	0% a 95% RH
Temperatura de armazenamento	-20°C a 85°C
Umidade do armazenamento	Não controlado para baixa temperatura. 90% RH em alta temperatura
Resfriamento	Extração de calor por ar forçado (ventilador)
Opções de montagem	Montagem em rack ou independente

Componente	Especificação
Cabo de alimentação	1,8 metros
Controle de potência	Botão de pressão
Redefinir	Interruptor momentâneo
Status e rede LEDs	LED RGB programável de 3 cores

O dispositivo apresenta Wi-Fi, Bluetooth e armazenamento para cartão SD, mas essas funcionalidades não podem ser usadas.

O AWS Panorama Appliance inclui dois parafusos para montagem em um rack de servidor. Você pode montar dois aparelhos side-by-side em um rack de 19 polegadas.

Cotas de serviço

O AWS Panorama aplica cotas aos recursos que você cria em sua conta e as aplicações que você implanta. Se você usa o AWS Panorama em várias AWS regiões, as cotas se aplicam separadamente a cada região. As cotas para AWS Panorama não são ajustáveis.

Os recursos no AWS Panorama incluem dispositivos, pacotes de nós de aplicações e instâncias de aplicações.

- Dispositivos: até 50 dispositivos registrados por Região.
- Pacotes de nós: 50 pacotes por Região, com até 20 versões por pacote.
- Instâncias de aplicações: até 10 aplicações por dispositivo. Cada aplicação pode monitorar até 8 streams de câmera. As implantações são limitadas a 200 por dia para cada dispositivo.

Quando você usa o AWS Panorama Application CLI ou AWS SDK com o serviço AWS Panorama, as cotas se aplicam ao número de chamadas de API que você faz. AWS Command Line Interface Você pode fazer até cinco solicitações no total por segundo. Um subconjunto de operações de API que criam ou modificam recursos aplica um limite adicional de 1 solicitação por segundo.

Para obter uma lista completa de cotas, acesse o console do Service Quotas ou consulte os endpoints e cotas do AWS Panorama no Referência geral da Amazon Web Services.

AWS Panorama permissões

Você pode usar o AWS Identity and Access Management (IAM) para gerenciar o acesso ao AWS Panorama serviço e aos recursos, como dispositivos e aplicativos. Para os usuários da sua conta que usam AWS Panorama, você gerencia as permissões em uma política de permissões que pode ser aplicada às funções do IAM. Para gerenciar as permissões de uma aplicação, crie um perfil e atribua-o à aplicação.

Para gerenciar permissões para usuários em sua conta, use a política gerenciada que AWS Panorama fornece ou crie sua própria política. Você precisa de permissões para outros AWS serviços para obter registros de aplicativos e equipamentos, visualizar métricas e atribuir uma função a um aplicativo.

Um AWS Panorama equipamento também tem uma função que lhe concede permissão para acessar AWS serviços e recursos. A função do equipamento é uma das funções de serviço que o AWS Panorama serviço usa para acessar outros serviços em seu nome.

Uma função de aplicativo é uma função de serviço separada que você cria para um aplicativo, para conceder a ele permissão para usar AWS serviços com AWS SDK for Python (Boto) o. Para criar um perfil de aplicação, você precisa de privilégios administrativos ou da ajuda de um administrador.

Restrinja permissões de usuário pelo recurso afetado por uma ação e, em alguns casos, por condições adicionais. Por exemplo, você pode especificar um padrão para o nome do recurso da Amazon (ARN) de uma função que exija que um usuário inclua o nome do usuário nos nomes das aplicações que ele cria. Para conhecer as condições e os recursos compatíveis com cada região, consulte Ações, recursos e chaves de condição AWS Panorama na Referência de autorização do serviço.

Para obter mais informações, consulte O que é o IAM? no Manual do usuário do IAM.

Tópicos

- Políticas do IAM baseadas em identidade para o AWS Panorama
- Perfis de serviço e recursos entre serviços do AWS Panorama
- Concessão de permissões a uma aplicação

Políticas do IAM baseadas em identidade para o AWS Panorama

Para conceder aos usuários da sua conta acesso ao Panorama AWS, você usa políticas baseadas em identidade no AWS Identity and Access Management (IAM). Aplique políticas baseadas em identidade aos perfis do IAM associados a um usuário. Também é possível conceder a usuários em outra conta permissão para assumir uma função na conta e acessar os recursos do AWS Panorama.

O AWS Panorama fornece políticas gerenciadas que concedem acesso a ações da API do AWS Panorama e, em alguns casos, acesso a outros serviços usados para desenvolver e gerenciar recursos do AWS Panorama. O AWS Panorama atualiza as políticas gerenciadas conforme necessário para garantir que os usuários tenham acesso a novos atributos quando eles forem lançados.

- AWSPanoramaFullAccess— Fornece acesso total ao AWS Panorama, aos pontos de acesso do AWS Panorama no Amazon S3, às credenciais do dispositivo e aos registros do dispositivo na AWS Secrets Manager Amazon. CloudWatch Inclui permissão para criar um perfil vinculado ao serviço para o AWS Panorama. Exibir política

A política `AWSPanoramaFullAccess` permite que você marque os recursos do AWS Panorama, mas não tem todas as permissões relacionadas à tag usadas pelo console do AWS Panorama. Para conceder essas permissões, adicione a seguinte política.

- ResourceGroupsandTagEditorFullAccess— Ver política

A política `AWSPanoramaFullAccess` não inclui permissão para comprar dispositivos no console do AWS Panorama. Para conceder essas permissões, adicione a seguinte política.

- ElementalAppliancesSoftwareFullAccess— Ver política

As políticas gerenciadas concedem permissão a ações da API sem restringir os recursos que um usuário pode modificar. Para um controle refinado, crie as próprias políticas que limitam o escopo das permissões de um usuário. Use a política de acesso total como ponto de partida para suas políticas.

> **Criar um perfil de serviço**
>
> No primeiro uso do console do AWS Panorama, você precisa de permissão para criar o perfil de serviço usado pelo AWS Panorama Appliance. Um perfil de serviço concede a um serviço permissão para gerenciar recursos ou interagir com outros serviços. Crie esse perfil antes de conceder acesso aos seus usuários.

Para obter detalhes sobre os recursos e condições que você pode usar para limitar o escopo das permissões de um usuário no AWS Panorama, consulte Ações, recursos e chaves de condição para o AWS Panorama na Referência de autorização do serviço.

Perfis de serviço e recursos entre serviços do AWS Panorama

O AWS Panorama usa outros serviços da AWS para gerenciar o AWS Panorama Appliance, armazenar dados e importar recursos de aplicações. Um perfil de serviço concede a um serviço permissão para gerenciar recursos ou interagir com outros serviços. Ao fazer login no console do AWS Panorama pela primeira vez, você cria os seguintes perfis de serviço:

- AWSServiceRoleForAWSPanorama— Permite que o AWS Panorama gerencie recursos no AWS IoT, no AWS Secrets Manager e no AWS Panorama.

  Política gerenciada: AWSPanoramaServiceLinkedRolePolicy
- AWSPanoramaApplianceServiceRole— Permite que um AWS Panorama Appliance carregue registros e obtenha objetos dos pontos de acesso do Amazon S3 criados pelo AWS Panorama. CloudWatch

  Política gerenciada: AWSPanoramaApplianceServiceRolePolicy

Para ver as permissões associadas a cada perfil, use o console do IAM. Sempre que possível, as permissões do perfil são restritas a recursos que correspondem a um padrão de nomenclatura usado pelo AWS Panorama. Por exemplo, `AWSServiceRoleForAWSPanorama` concede somente permissão para que o serviço acesse AWS IoT recursos que tenham `panorama` em seu nome.

Seções

Proteção do perfil do dispositivo

O AWS Panorama Appliance usa o perfil `AWSPanoramaApplianceServiceRole` para acessar recursos em sua conta. O dispositivo tem permissão para fazer upload de CloudWatch registros para Logs, ler credenciais de transmissão de câmera e acessar artefatos de AWS Secrets Manager aplicativos nos pontos de acesso do Amazon Simple Storage Service (Amazon S3) criados pelo AWS Panorama.

> **Note**
>
> As aplicações não usam as permissões do dispositivo. Para dar permissão à sua aplicação para usar serviços da AWS , crie um perfil de aplicação.

O AWS Panorama usa o mesmo perfil de serviço com todos os dispositivos em sua conta e não usa perfis entre contas. Para obter uma camada adicional de segurança, você pode modificar a política de confiança do perfil do dispositivo para impor isso explicitamente, o que é uma prática recomendada quando você usa perfis para conceder permissão a um serviço para acessar recursos em sua conta.

Para atualizar a política de confiança do dispositivo

1. Abra a função do dispositivo no console do IAM: AWSPanoramaApplianceServiceRole
2. Selecione Edit trust relationship (Editar relação de confiança).
3. Atualize o conteúdo da política e escolha Atualizar política de confiança.

A política de confiança a seguir inclui uma condição que garante que, quando o AWS Panorama assumir o perfil de dispositivo, ele faça isso para um dispositivo em sua conta. A condição `aws:SourceAccount` compara o ID da conta especificado pelo AWS Panorama ao ID que você inclui na política.

Example política de confiança: conta específica

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

```
  ]
}
```

Se você quiser restringir ainda mais o AWS Panorama e só permitir que ele assuma o perfil com um dispositivo específico, você pode especificar o dispositivo pelo ARN. A condição `aws:SourceArn` compara o ARN do dispositivo especificado pelo AWS Panorama ao ID que você inclui na política.

Example política de confiança: 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"
        }
      }
    }
  ]
}
```

Se você redefinir e reprovisionar o dispositivo, deverá remover temporariamente a condição do ARN de origem e depois adicioná-la novamente com o novo ID do dispositivo.

Para obter mais informações sobre essas condições e as melhores práticas de segurança quando os serviços usam perfis para acessar recursos em sua conta, consulte O problema de "confused deputy" no Guia do usuário do IAM.

Uso de outros serviços

O AWS Panorama cria ou acessa recursos nos seguintes serviços:

- AWS IoT: coisas, políticas, certificados e trabalhos para o AWS Panorama Appliance
- Amazon S3: pontos de acesso para preparar modelos, códigos e configurações de aplicações.
- Secrets Manager: credenciais de curto prazo para o AWS Panorama Appliance.

Para obter informações sobre o formato do nome do recurso da Amazon (ARN) ou os escopos de permissão para cada serviço, consulte os tópicos no Guia do usuário do IAM que estão vinculados a esta lista.

Concessão de permissões a uma aplicação

Você pode criar uma função para seu aplicativo para conceder permissão para chamar AWS serviços. Por padrão, as aplicações não têm nenhuma permissão. Você cria um perfil de aplicação no IAM e o atribui a uma aplicação durante a implantação. Para conceder à sua aplicação somente as permissões necessárias, crie um perfil para ela com permissões para ações específicas da API.

O aplicativo de amostra inclui um AWS CloudFormation modelo e um script que criam uma função de aplicativo. Trata-se de um perfil de serviço que o AWS Panorama pode assumir. Essa função concede permissão para que o aplicativo chame CloudWatch para fazer upload de métricas.

Example aws-panorama-sample.yml — Função do aplicativo

```
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/
```

Você pode estender esse script para conceder permissões a outros serviços, especificando uma lista de ações ou padrões de API para o valor de `Action`.

Para mais informações sobre permissões no AWS Panorama, consulte AWS Panorama permissões.

Gerenciando o AWS Panorama equipamento

O AWS Panorama equipamento é o hardware que executa seus aplicativos. Você usa o AWS Panorama console para registrar um equipamento, atualizar seu software e implantar aplicativos nele. O software do AWS Panorama aparelho se conecta às transmissões da câmera, envia quadros de vídeo para seu aplicativo e exibe a saída de vídeo em um monitor conectado.

Depois de configurar seu dispositivo ou outro dispositivo compatível, você registra as câmeras para uso com aplicações. Você gerencia os fluxos de câmera no AWS Panorama console. Ao implantar uma aplicação, você escolhe quais streams de câmera o dispositivo envia para processamento.

Para tutoriais que apresentam o AWS Panorama Appliance com um aplicativo de amostra, consulte. Começando com AWS Panorama

Tópicos

Gerenciamento de um AWS Panorama Appliance

Use o console do AWS Panorama para configurar, atualizar ou cancelar o registro do AWS Panorama Appliance e de outros dispositivos compatíveis.

Para configurar um dispositivo, siga as instruções no tutorial de conceitos básicos. O processo de configuração cria os recursos no AWS Panorama que rastreiam seu dispositivo e coordenam atualizações e implantações.

Para registrar um dispositivo com a API do AWS Panorama, consulte Automatização do registro de dispositivos.

Seções

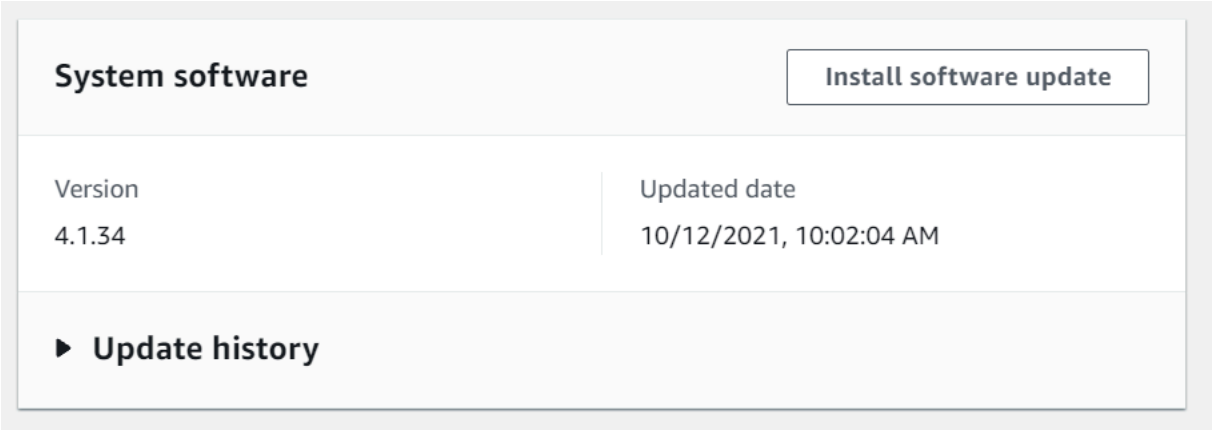
- Atualize o software do dispositivo
- Cancelamento do registro de um dispositivo
- Reinicialização de um dispositivo
- Redefinição de um dispositivo

Atualize o software do dispositivo

Visualize e implante atualizações de software no dispositivo usando o console do AWS Panorama. As atualizações podem ser obrigatórias ou opcionais. Quando uma atualização necessária está disponível, o console solicita que você a aplique. Você pode aplicar atualizações opcionais na página Configurações do dispositivo.

Para atualizar o software do dispositivo

1. Abra a página Dispositivos do console do AWS Panorama.
2. Escolha um dispositivo.
3. Escolha Configurações
4. Em Software do sistema, escolha Instalar atualização de software.



5. Escolha uma nova versão e, em seguida, selecione Instalar.

Cancelamento do registro de um dispositivo

Se você terminar de trabalhar com um dispositivo, poderá usar o console do AWS Panorama para cancelar o registro e excluir os recursos associados. AWS IoT

Para excluir um dispositivo

1. Abra a página Dispositivos do console do AWS Panorama.
2. Escolha o nome do dispositivo.
3. Escolha Excluir.
4. Insira o nome do dispositivo e escolha Excluir.

Quando você exclui um dispositivo do serviço AWS Panorama, os dados no dispositivo não são excluídos automaticamente. Um equipamento com registro cancelado não pode se conectar aos AWS serviços e não pode ser registrado novamente até que seja redefinido.

Reinicialização de um dispositivo

Você pode reinicializar um dispositivo remotamente.

Para reinicializar um dispositivo

1. Abra a página Dispositivos do console do AWS Panorama.
2. Escolha o nome do dispositivo.
3. Escolha Reboot.

O console envia uma mensagem ao dispositivo para reinicializá-lo. Para receber o sinal, o dispositivo deve ser capaz de se conectar ao AWS IoT. Para reinicializar um dispositivo com a API do AWS Panorama, consulte Reinicialização de dispositivos.

Redefinição de um dispositivo

Para usar um dispositivo em uma região diferente ou com uma conta diferente, redefina-o e reprovisione-o com um novo certificado. A redefinição do dispositivo aplica a versão mais recente do software necessária e exclui todos os dados da conta.

Para iniciar uma operação de redefinição, o dispositivo deve estar conectado e desligado. Pressione e segure os botões ligar/desligar e redefinir por cinco segundos. Quando você solta os botões, a luz de status pisca em laranja. Espere até que a luz de status pisque em verde antes de provisionar ou desconectar o dispositivo.

Você também pode redefinir o software do dispositivo sem excluir certificados do dispositivo. Para obter mais informações, consulte Botões Ligar/Desligar e Redefinir.

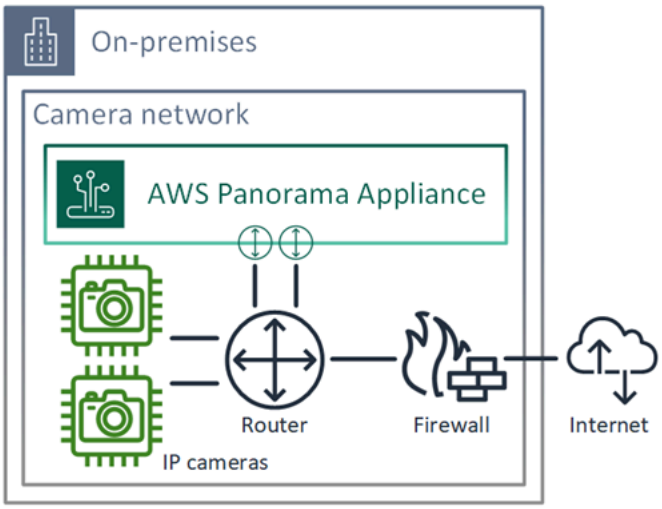
Conectar o AWS Panorama Appliance à sua rede

O AWS Panorama Appliance exige conectividade com a AWS nuvem e com sua rede local de câmeras IP. Você pode conectar o dispositivo a um único firewall que conceda acesso a ambos ou conectar cada uma das duas interfaces de rede do dispositivo a uma sub-rede diferente. Em ambos os casos, você deve proteger as conexões de rede do dispositivo para impedir o acesso não autorizado aos streams da câmera.

Seções

Configuração de rede única

O dispositivo tem duas portas Ethernet. Se você rotear todo o tráfego emitido e recebido pelo dispositivo usando um único roteador, poderá usar a segunda porta para redundância, caso a conexão física com a primeira porta seja interrompida. Configure seu roteador para permitir que o dispositivo se conecte somente aos streams de câmeras e à Internet e para impedir que os streams de câmeras saiam da sua rede interna.

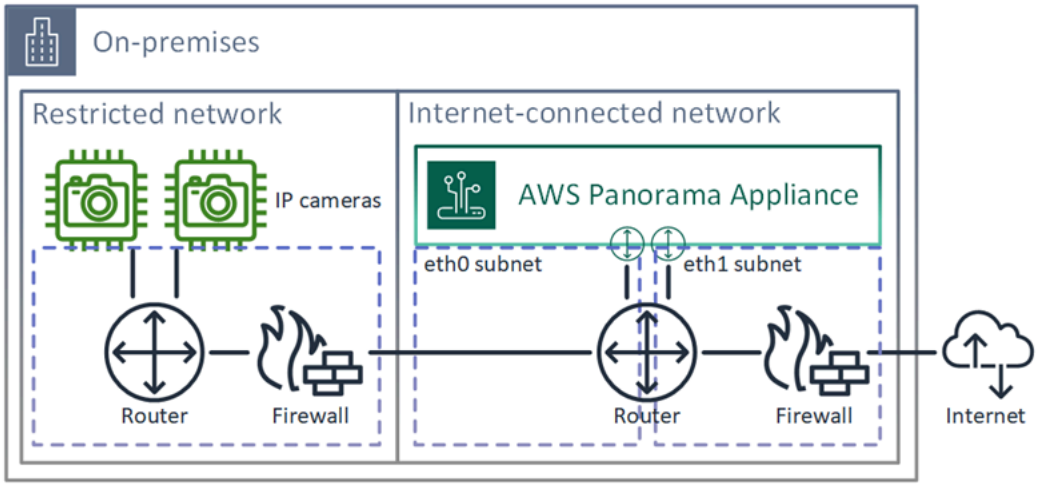


Para obter detalhes sobre as portas e endpoints aos quais o dispositivo precisa de acesso, consulte Configurar o acesso a serviço e Configuração do acesso à rede local.

Configuração de rede dupla

Para obter uma camada extra de segurança, você pode colocar o dispositivo em uma rede conectada à Internet separada da sua rede de câmeras. Um firewall entre sua rede restrita de câmeras e a rede do dispositivo só permite que o dispositivo acesse streams de vídeo. Se antes sua rede de câmeras estava isolada por motivos de segurança, talvez você prefira esse método em vez de conectar a rede de câmeras a um roteador que também conceda acesso à Internet.

O exemplo a seguir mostra o dispositivo se conectando a uma sub-rede diferente em cada porta. O roteador coloca a interface `eth0` em uma sub-rede que é roteada para a rede da câmera, e `eth1` em uma sub-rede que roteia para a Internet.



Você pode confirmar o endereço IP e o endereço MAC de cada porta no console do AWS Panorama.

Configurar o acesso a serviço

Durante o provisionamento, você pode configurar o dispositivo para solicitar um endereço IP específico. Escolha um endereço IP com antecedência para simplificar a configuração do firewall e garantir que o endereço do dispositivo não mude se ele ficar off-line por um longo período de tempo.

O equipamento usa AWS serviços para coordenar atualizações e implantações de software. Configure seu firewall para permitir que o dispositivo se conecte a esses endpoints.

Acesso à Internet

- AWS IoT (HTTPS e MQTT, portas 443, 8443 e 8883) — AWS IoT Core e endpoints de gerenciamento de dispositivos. Para obter detalhes, consulte Endpoints e cotas do AWS IoT Device Management na Referência geral da Amazon Web Services.
- AWS IoT credenciais (HTTPS, porta 443) — `credentials.iot.<region>.amazonaws.com` e subdomínios.
- Amazon Elastic Container Registry (HTTPS, porta 443): `api.ecr.<region>.amazonaws.com`, `dkr.ecr.<region>.amazonaws.com` e subdomínios.
- Amazon CloudWatch (HTTPS, porta 443) —`monitoring.<region>.amazonaws.com`.
- Amazon CloudWatch Logs (HTTPS, porta 443) —`logs.<region>.amazonaws.com`.
- Amazon Simple Storage Service (HTTPS, porta 443): `s3.<region>.amazonaws.com`, `s3-accesspoint.<region>.amazonaws.com` e subdomínios.

Se seu aplicativo chamar outros AWS serviços, o equipamento também precisará acessar os endpoints desses serviços. Para obter mais informações, consulte Endpoints e cotas de serviços.

Configuração do acesso à rede local

O dispositivo precisa acessar os streams de vídeo RTSP localmente, mas não pela Internet. Configure seu firewall para permitir que o dispositivo acesse streams RTSP na porta 554 internamente e para não permitir que streams entrem ou saiam pela Internet.

Acesso local

- Protocolo de streaming em tempo real (RTSP, porta 554): para ler streams de câmeras.
- Protocolo de tempo de rede (NTP, porta 123): para manter o relógio do dispositivo sincronizado. Se você não executa um servidor NTP em sua rede, o dispositivo também pode se conectar a servidores NTP públicos pela Internet.

Conectividade privada

O AWS Panorama Appliance não precisa de acesso à Internet se você o implantar em uma sub-rede VPC privada com uma conexão VPN a. AWS Você pode usar Site-to-Site VPN ou AWS Direct Connect criar uma conexão VPN entre um roteador local e. AWS Na sua sub-rede VPC privada, você cria endpoints que permitem que o dispositivo se conecte ao Amazon Simple Storage Service AWS

IoT e a outros serviços. Para obter mais informações, consulte Conexão de um dispositivo a uma sub-rede privada.

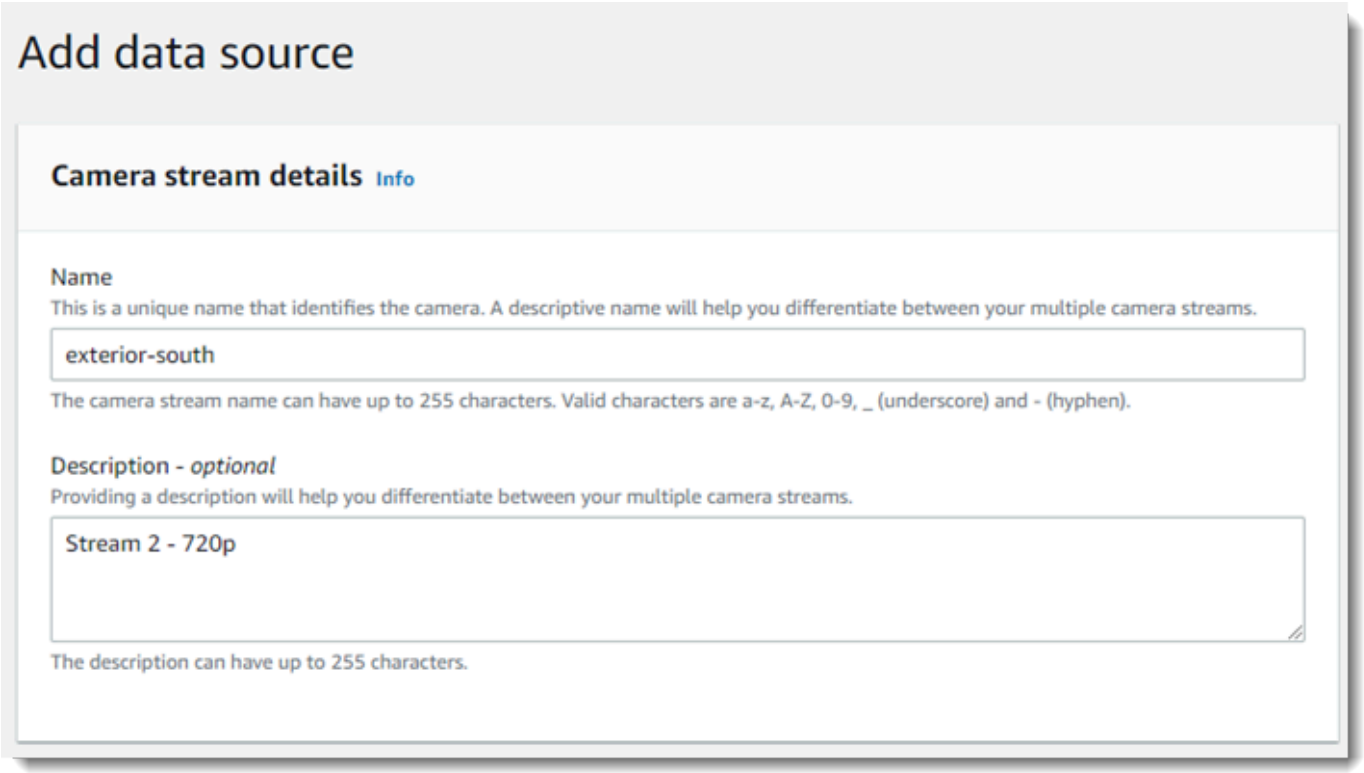
Gerenciamento de streams de câmeras no AWS Panorama

Para registrar streams de vídeo como fontes de dados para sua aplicação, use o console do AWS Panorama. Uma aplicação pode processar vários streams simultaneamente, e vários dispositivos podem se conectar ao mesmo stream.

> **Important**
>
> Uma aplicação pode se conectar a qualquer stream de câmera que seja roteável da rede local à qual está conectado. Para proteger seus streams de vídeo, configure sua rede para permitir somente tráfego RTSP localmente. Para obter mais informações, consulte Segurança no AWS Panorama.

Para registrar um stream de câmera

1. Abra a Página de fontes de dados do console do AWS Panorama.
2. Escolha Adicionar fonte de dados.



3. Configure as definições a seguir.

- Nome: um nome para o stream da câmera.
- Descrição: uma breve descrição da câmera, sua localização ou outros detalhes.
- URL RTSP: URL que especifica o endereço IP da câmera e o caminho para o stream. Por exemplo, `rtsp://192.168.0.77/live/mpeg4/`
- Credenciais: se o stream da câmera estiver protegido por senha, especifique o nome de usuário e a senha.

4. Escolha Salvar.

Para registrar um stream de câmera com a API do AWS Panorama, consulte Automatização do registro de dispositivos.

Para obter uma lista de câmeras compatíveis com o AWS Panorama Appliance, consulte Modelos e câmeras de visão computacional compatíveis.

Remoção de um stream

Você pode excluir um stream de câmera no console do AWS Panorama.

Para remover um stream de câmera

1. Abra a Página de fontes de dados do console do AWS Panorama.
2. Escolha um stream de câmera.
3. Escolha Excluir fonte de dados.

A remoção de um stream de câmera do serviço não interrompe a execução de aplicações nem exclui as credenciais da câmera do Secrets Manager. Para excluir segredos, use o console do Secrets Manager.

Gerenciamento de aplicações em um AWS Panorama Appliance

Uma aplicação é uma combinação de código, modelos e configuração. Na página Dispositivos no console do AWS Panorama, você pode gerenciar aplicações no dispositivo.

Para gerenciar aplicações em um AWS Panorama Appliance

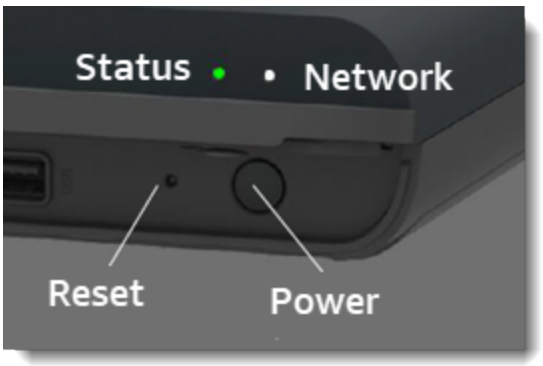
1. Abra a página Dispositivos do console do AWS Panorama.
2. Escolha um dispositivo.

A página Aplicações implantadas mostra as aplicações que foram implantadas no dispositivo.

Use as opções desta página para remover aplicações implantadas do dispositivo ou substituir uma aplicação em execução por uma nova versão. Você também pode clonar uma aplicação (em execução ou excluída) para implantar uma nova cópia dela.

Botões e luzes do AWS Panorama Appliance

O AWS Panorama Appliance tem duas luzes LED acima do botão Ligar/Desligar que indicam o status do dispositivo e a conectividade da rede.



Luz de status

A LEDs mudança de cor e pisca para indicar o status. Um piscar lento ocorre uma vez a cada três segundos. Um piscar rápido ocorre uma vez por segundo.

Estados do LED de status

- Verde piscando rapidamente: o dispositivo está inicializando.
- Verde sólido: o dispositivo está funcionando normalmente.
- Azul piscando lentamente — O equipamento está copiando arquivos de configuração e tentando se registrar no. AWS IoT
- Azul piscando rapidamente: o dispositivo está copiando uma imagem de log para uma unidade USB.
- Vermelho piscando rapidamente: o dispositivo encontrou um erro durante o startup ou está superaquecido.
- Laranja piscando lentamente: o dispositivo está restaurando a versão mais recente do software.
- Laranja piscando rapidamente: o dispositivo está restaurando a versão mínima do software.

Luz de rede

O LED da rede tem os seguintes estados:

Estados do LED de rede

- Verde sólido: um cabo Ethernet está conectado.
- Verde piscando: o dispositivo está se comunicando pela rede.
- Vermelho sólido: um cabo Ethernet não está conectado.

Botões Ligar/Desligar e Redefinir

Os botões Ligar/Desligar e Redefinir estão na parte frontal do dispositivo, embaixo de uma capa protetora. O botão Redefinir é menor e embutido. Use uma pequena chave de fenda ou clipe de papel para pressioná-lo.

Para redefinir um dispositivo

1. O dispositivo deve estar conectado e desligado. Para desligar o dispositivo, segure o botão Ligar/Desligar por 1 segundo e aguarde a conclusão da sequência de desligamento. A sequência de desligamento leva cerca de 10 segundos.
2. Para redefinir o dispositivo, use a seguinte combinação de botões. Um toque curto dura 1 segundo. Um toque longo dura 5 segundos. Para operações que exigem vários botões, pressione e segure os dois botões simultaneamente.

   - Redefinição completa: pressione demoradamente os botões Ligar/Desligar e Redefinir.

     Restaura a versão mínima do software e exclui todos os arquivos de configuração e aplicações.
   - Restaurar a versão mais recente do software: pressione brevemente o botão Redefinir.

     Reaplica a atualização de software mais recente ao dispositivo.
   - Restaurar a versão mínima do software: pressione demoradamente o botão Redefinir.

     Reaplica a atualização de software mais recente necessária ao dispositivo.
3. Solte os dois botões. O dispositivo é ligado, e a luz de status pisca em laranja por vários minutos.
4. Quando o dispositivo estiver pronto, a luz de status piscará em verde.

A redefinição de um dispositivo não o exclui do serviço do AWS Panorama. Para obter mais informações, consulte Cancelamento do registro de um dispositivo.

Gerenciando AWS Panorama aplicativos

Os aplicativos são executados no AWS Panorama equipamento para realizar tarefas de visão computacional em streams de vídeo. Você pode criar aplicativos de visão computacional combinando código Python e modelos de aprendizado de máquina e implantá-los no AWS Panorama Appliance pela Internet. As aplicações podem enviar vídeo para um monitor ou usar o SDK da AWS para enviar resultados para os serviços da AWS.

Tópicos

Implantar uma aplicação

Para implantar uma aplicação, use a CLI da aplicação do AWS Panorama, importe-a para sua conta, crie o contêiner, faça upload e registre os ativos e crie uma instância da aplicação. Este tópico aborda cada uma dessas etapas em detalhes e descreve o que acontece em segundo plano.

Se você ainda não implantou uma aplicação, consulte uma explicação passo a passo em Começando com AWS Panorama.

Para obter mais informações sobre como personalizar e estender a aplicação de exemplo, consulte Criação de AWS Panorama aplicativos.

Seções

Instale a CLI da aplicação do AWS Panorama

Para instalar a CLI do aplicativo AWS Panorama e use AWS CLI pip.

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

Para criar imagens de aplicações com a CLI da aplicação do AWS Panorama, você precisa do Docker. No Linux, bibliotecas `qemu` e bibliotecas de sistema relacionadas também são necessárias. Para obter mais informações sobre como instalar e configurar a CLI do aplicativo AWS Panorama, consulte o arquivo README no repositório do projeto. GitHub

- github. com/aws/aws-clipe panorâmico

Para obter instruções sobre como configurar um ambiente de compilação no Windows com WSL2, consulteConfigurar um ambiente de desenvolvimento no Windows.

Importação de uma aplicação

Se você estiver trabalhando com uma aplicação de exemplo ou uma aplicação fornecida por terceiros, use a CLI da aplicação do AWS Panorama para importar a aplicação.

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

Esse comando renomeia os pacotes de aplicação com o ID da sua conta. Os nomes dos pacotes começam com o ID da conta na qual foram implantados. Ao implantar uma aplicação em várias contas, você deve importar e empacotar a aplicação separadamente para cada conta.

Por exemplo, a aplicação de exemplo deste guia é um pacote de código e um pacote de modelo, cada um nomeado com um ID de conta reservado. O `import-application` comando os renomeia para usar o ID da conta que a CLI infere das credenciais do seu espaço de trabalho. AWS

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

`123456789012` é substituído pelo ID da sua conta nos nomes dos diretórios de pacotes e no manifesto da aplicação (`graph.json`), que se refere a eles. Você pode confirmar o ID da sua conta ligando `aws sts get-caller-identity` com AWS CLI o.

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

```
}
```

Criar uma imagem de contêiner

O código da aplicação é empacotado em uma imagem de contêiner do Docker, que inclui o código da aplicação e as bibliotecas que você instala no seu Dockerfile. Use o comando `build-container` da CLI da aplicação do AWS Panorama para criar uma imagem do Docker e exportar uma imagem do sistema de arquivos.

```
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
```

Esse comando cria uma imagem do Docker chamada `code_asset` e exporta um sistema de arquivos para um arquivo `.tar.gz` na pasta `assets`. A CLI extrai a imagem base da aplicação do Amazon Elastic Container Registry (Amazon ECR), conforme especificado no Dockerfile da aplicação.

Além do arquivo do contêiner, a CLI cria um ativo para o descritor do pacote (`descriptor.json`). Ambos os arquivos são renomeados com um identificador exclusivo que reflete um hash do arquivo original. A CLI da aplicação do AWS Panorama também adiciona um bloco à configuração do pacote, que registra os nomes dos dois ativos. Esses nomes são usados pelo dispositivo durante o processo de implantação.

Example packages/123456789012-SAMPLE_CODE-1.0/package.json: com bloco de ativos

```
{
    "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"
                    },
```

O nome do ativo de código, especificado no comando `build-container`, deve corresponder ao valor do campo `asset` na configuração do pacote. No exemplo anterior, os dois valores são `code_asset`.

Importação de um modelo

Sua aplicação pode ter um arquivo de modelo na pasta de ativos, ou você pode baixar um arquivo de modelo separadamente. Se você tiver um novo modelo, um modelo atualizado ou um arquivo descritor de modelo atualizado, use o comando `add-raw-model` para importá-lo.

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

```
    --packages-path packages/210987654321-SQUEEZENET_PYTORCH-1.0
```

Se você precisar apenas atualizar o arquivo descritor, poderá reutilizar o modelo existente no diretório de ativos. Talvez seja necessário atualizar o arquivo descritor para configurar atributos, como o modo de precisão de ponto flutuante. Por exemplo, o script a seguir mostra como fazer isso com a aplicação de exemplo.

Example scripts utilitários/ .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
```

As alterações no arquivo descritor no diretório do pacote do modelo não serão aplicadas até que você o reimporte com a CLI. A CLI atualiza a configuração do pacote do modelo com os novos nomes de ativos no local, da mesma forma como atualiza a configuração do pacote de código da aplicação quando você reconstrói um contêiner.

Upload de ativos da aplicação

Para fazer upload e registrar os ativos da aplicação, que incluem o arquivo do modelo, o arquivo do sistema de arquivos do contêiner e os respectivos arquivos descritores, use o comando `package-application`.

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

```
panorama-210987654321-6k75xmpl2jypelgzst7uux62ye/210987654321/nodePackages/
SQUEEZENET_PYTORCH/
binaries/87fbxmpl6f18aeae4d1e3ff8bbc6147390feaf47d85b5da34f8374974ecc4aaf.json
Called register package version for SQUEEZENET_PYTORCH with patch version
 5d3cxmplb7113faa1d130f97f619655d8ca12787c751851a0e155e50eb5e3e96
...
```

Se não houver alterações em um arquivo de ativo ou na configuração do pacote, a CLI os ignorará.

```
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
```

A CLI carrega os ativos de cada pacote em um Ponto de Acesso Amazon S3 que é específico da sua conta. O AWS Panorama gerencia o ponto de acesso para você e fornece informações sobre ele por meio da DescribePackageAPI. A CLI carrega os ativos de cada pacote no local fornecido para esse pacote e os registra no serviço AWS Panorama com as configurações descritas na configuração do pacote.

Implantação de uma aplicação com o console do AWS Panorama

Você pode implantar uma aplicação com o console do AWS Panorama. Durante o processo de implantação, você escolhe quais streams de câmera passar para o código da aplicação e configura as opções fornecidas pelo desenvolvedor da aplicação.

Para implantar uma aplicação

1. Abra a Página de aplicações implantadas do console do AWS Panorama.
2. Escolha Implantar aplicação.
3. Cole o conteúdo do manifesto da aplicação, `graph.json`, no editor de texto. Escolha Próximo.
4. Digite um nome e uma descrição.
5. Escolha Prosseguir para a implantação.
6. Escolha Iniciar implantação.
7. Se a aplicação usar uma função, escolha-a no menu suspenso. Escolha Próximo.

8. Escolha Selecionar dispositivo e, em seguida, escolha seu dispositivo. Escolha Próximo.
9. Na etapa Selecionar fontes de dados, escolha Visualizar entrada(s) e adicione o stream da câmera como uma fonte de dados. Escolha Próximo.
10. Na etapa Configurar, realize todas as configurações específicas da aplicação definidas pelo desenvolvedor. Escolha Próximo.
11. Escolha Implantar e Concluído.
12. Na lista de aplicações implantadas, escolha a aplicação para monitorar seu status.

O processo de implantação leva de 15 a 20 minutos. A saída do dispositivo pode ficar em branco por um longo período enquanto a aplicação é iniciada. Se for exibido um erro, consulte Solução de problemas.

Automatização da implantação da aplicação

Você pode automatizar o processo de implantação do aplicativo com a CreateApplicationInstanceAPI. A API usa dois arquivos de configuração como entrada. O manifesto da aplicação especifica os pacotes usados e seus relacionamentos. O segundo arquivo é um arquivo de substituições que especifica as substituições de valores no manifesto da aplicação no momento da implantação. O uso de um arquivo de substituições permite que você use o mesmo manifesto da aplicação para implantar a aplicação com diferentes streams de câmera e defina outras configurações específicas da aplicação.

Para obter mais informações e exemplos de scripts para cada uma das etapas deste tópico, consulte Automatização da implantação da aplicação.

Gerenciar aplicações no console do AWS Panorama

Use o console do AWS Panorama para gerenciar aplicações implantadas.

Seções

Atualização ou cópia de uma aplicação

Para atualizar uma aplicação, use a opção Substituir. Ao substituir uma aplicação, você pode atualizar o código ou os modelos dela.

Para atualizar uma aplicação

1. Abra a Página de aplicações implantadas do console do AWS Panorama.
2. Escolha a aplicação.
3. Selecione Replace (Substituir).
4. Siga as instruções para criar uma nova versão ou aplicação.

Há também uma opção Clonar que funciona de forma semelhante à opção Substituir, mas não remove a versão antiga da aplicação. Você pode usar essa opção para testar alterações em uma aplicação sem interromper a versão em execução ou para reimplantar uma versão que você já excluiu.

Excluir versões e aplicações

Para remover as versões não utilizadas da aplicação, exclua-as dos seus dispositivos.

Para excluir uma aplicação

1. Abra a Página de aplicações implantadas do console do AWS Panorama.
2. Escolha a aplicação.
3. Escolha Excluir do dispositivo.

Configuração do pacote

Quando você usa o comando `panorama-cli package-application` da CLI do AWS Panorama Application, a CLI carrega os ativos da sua aplicação no Amazon S3 e os registra no AWS Panorama. Os ativos incluem arquivos binários (imagens e modelos de contêineres) e arquivos descritores, que o AWS Panorama Appliance baixa durante a implantação. Para registrar os ativos de um pacote, você fornece um arquivo de configuração de pacote separado que define o pacote, seus ativos e sua interface.

O exemplo a seguir mostra uma configuração de pacote para um nó de código com uma entrada e uma saída. A entrada de vídeo fornece acesso aos dados de imagem de um stream de câmera. O nó de saída envia imagens processadas para um monitor.

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"
                    }
                ]
            }
        ]
    }
}
```

A seção `assets` especifica os nomes dos artefatos que a CLI do AWS Panorama Application enviou para o Amazon S3. Se você importar uma aplicação de exemplo ou uma aplicação de outro usuário, essa seção poderá estar vazia ou se referir a ativos que não estão na sua conta. Quando você executa `panorama-cli package-application`, a CLI do AWS Panorama Application preenche essa seção com os valores corretos.

O manifesto da aplicação do AWS Panorama

Ao implantar uma aplicação, você fornece um arquivo de configuração chamado manifesto da aplicação. Esse arquivo define a aplicação como um gráfico com nós e bordas. O manifesto da aplicação faz parte do código-fonte da aplicação e é armazenado no diretório `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"
            }
        ]
    }
}
```

Os nós são conectados por bordas, que especificam mapeamentos entre as entradas e saídas dos nós. A saída de um nó se conecta à entrada de outro, formando um gráfico.

Esquema JSON

O formato do manifesto da aplicação e dos documentos de substituição é definido em um esquema JSON. Você pode usar o esquema JSON para validar seus documentos de configuração antes da implantação. O esquema JSON está disponível no repositório deste guia. GitHub

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

Nós da aplicação

Os nós são modelos, código, streams de câmera, saída e parâmetros. Um nó tem uma interface, que define suas entradas e saídas. A interface pode ser definida em um pacote na sua conta, em um pacote fornecido pelo AWS Panorama ou em um tipo incorporado.

No exemplo a seguir, `code_node` e `model_node` referem-se aos pacotes de exemplo de código e modelo incluídos na aplicação de exemplo. `camera_node` usa um pacote fornecido pelo AWS Panorama para criar um espaço reservado para um stream de câmera que você especifica durante a implantação.

Example graph.json: nós

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

Edges (Bordas)

As bordas mapeiam a saída de um nó para a entrada de outro. No exemplo a seguir, a primeira borda mapeia a saída de um nó de stream de câmera para a entrada de um nó de código da aplicação. Os nomes `video_in` e `video_out` são definidos nas interfaces dos pacotes de nós.

Example graph.json: bordas

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

No código da aplicação, use os atributos `inputs` e `outputs` para obter imagens do stream de entrada e enviar imagens para o stream de saída.

Example application.py: entrada e saída 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)
```

Nós abstratos

Em um manifesto de aplicação, um nó abstrato se refere a um pacote definido pelo AWS Panorama, que você pode usar como espaço reservado no manifesto da sua aplicação. O AWS Panorama fornece dois tipos de nós abstratos.

- Stream de câmera: escolha o stream de câmera que a aplicação usa durante a implantação.

  Nome do pacote: `panorama::abstract_rtsp_media_source`

  Nome da interface: `rtsp_v1_interface`

- Saída HDMI: indica que a aplicação envia vídeo.

  Nome do pacote: `panorama::hdmi_data_sink`

  Nome da interface: `hdmi0`

O exemplo a seguir mostra um conjunto básico de pacotes, nós e bordas para uma aplicação que processa streams de câmera e envia vídeo para um monitor. O nó da câmera, que usa a interface do pacote `abstract_rtsp_media_source` no AWS Panorama, pode aceitar vários streams de câmera como entrada. O nó de saída, que faz referência a `hdmi_data_sink`, dá ao código da aplicação acesso a um buffer de vídeo que é enviado pela porta HDMI do dispositivo.

Example graph.json: nós abstratos

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

Parâmetros da aplicação

Os parâmetros são nós de tipo básico que podem ser substituídos durante a implantação. Um parâmetro pode ter um valor padrão e um decorador, que instrui o usuário da aplicação sobre como configurá-la.

Tipos de parâmetros

- `string`: uma string. Por exemplo, `DEBUG`.
- `int32`: um valor inteiro. Por exemplo, `20`
- `float32`: um número de ponto flutuante. Por exemplo, `47.5`
- `boolean`: `true` ou `false`.

O exemplo a seguir mostra dois parâmetros, uma string e um número, que são enviados para um nó de código como entradas.

Example graph.json: parâmetros

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

Você pode modificar os parâmetros diretamente no manifesto da aplicação ou fornecer novos valores no momento da implantação com substituições. Para obter mais informações, consulte Configuração de tempo de implantação com substituições.

Configuração de tempo de implantação com substituições

Você configura parâmetros e nós abstratos durante a implantação. Se você usar o console do AWS Panorama para implantar, poderá especificar um valor para cada parâmetro e escolher um stream de câmera como entrada. Se você usar a API do AWS Panorama para implantar aplicações, especifique essas configurações com um documento de substituição.

Um documento de substituição tem estrutura semelhante a um manifesto de aplicação. Para parâmetros com tipos básicos, defina um nó. Para streams de câmera, defina um nó e um pacote que mapeia para um stream de câmeras registrado. Em seguida, defina uma substituição para cada nó que especifica o nó do manifesto da aplicação que ele substitui.

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"
    }
}
```

No exemplo anterior, o documento define substituições para um parâmetro de string e um nó de câmera abstrato. Isso `node0verrides` informa ao AWS Panorama quais nós neste documento substituem quais nós no manifesto da aplicação.

Criação de AWS Panorama aplicativos

Os aplicativos são executados no AWS Panorama equipamento para realizar tarefas de visão computacional em streams de vídeo. Você pode criar aplicativos de visão computacional combinando código Python e modelos de aprendizado de máquina e implantá-los no AWS Panorama Appliance pela Internet. As aplicações podem enviar vídeo para um monitor ou usar o SDK da AWS para enviar resultados para os serviços da AWS.

Um modelo analisa imagens para detectar pessoas, veículos e outros objetos. Com base nas imagens que viu durante o treinamento, o modelo diz o que ele acha que algo é, e o nível de confiança dessa suposição. Você pode treinar modelos com seus próprios dados de imagem ou começar com um exemplo.

O código da aplicação processa imagens estáticas de um stream de câmera, as envia para um modelo e processa o resultado. Um modelo pode detectar vários objetos e retornar suas formas e localização. O código pode usar essas informações para adicionar texto ou gráficos ao vídeo ou para enviar resultados a um serviço da AWS para armazenamento ou processamento adicional.

Para obter imagens de um stream, interagir com um modelo e gerar vídeo, o código do aplicativo usa o SDK do AWS Panorama aplicativo. O SDK do aplicativo é uma biblioteca Python que oferece suporte a modelos gerados PyTorch com, MXNet Apache e. TensorFlow

Tópicos

Modelos de visão computacional

Um modelo de visão computacional é um programa de software treinado para detectar objetos em imagens. Um modelo aprende a reconhecer um conjunto de objetos analisando primeiro as imagens desses objetos por meio de treinamento. Um modelo de visão computacional usa uma imagem como entrada e gera informações sobre os objetos que detecta, como tipo de objeto e sua localização. O AWS Panorama oferece suporte a modelos de visão computacional criados com PyTorch MXNet, Apache e. TensorFlow

> **Note**
>
> Para obter uma lista de modelos pré-criados que foram testados com o AWS Panorama, consulte Compatibilidade de modelos.

Seções

Uso de modelos em código

Um modelo retorna um ou mais resultados, que podem incluir probabilidades de classes detectadas, informações de localização e outros dados. O exemplo a seguir mostra como executar inferência em uma imagem de um stream de vídeo e enviar a saída do modelo para uma função de processamento.

Example application.py: inferência

```
    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)
```

O exemplo a seguir mostra uma função que processa resultados do modelo de classificação básico. O modelo de amostra retorna uma matriz de probabilidades, que é o primeiro e único valor na matriz de resultados.

Example application.py: resultados do processo

```
    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)
```

O código da aplicação encontra os valores com as maiores probabilidades e os mapeia para rótulos em um arquivo de recurso que é carregado durante a inicialização.

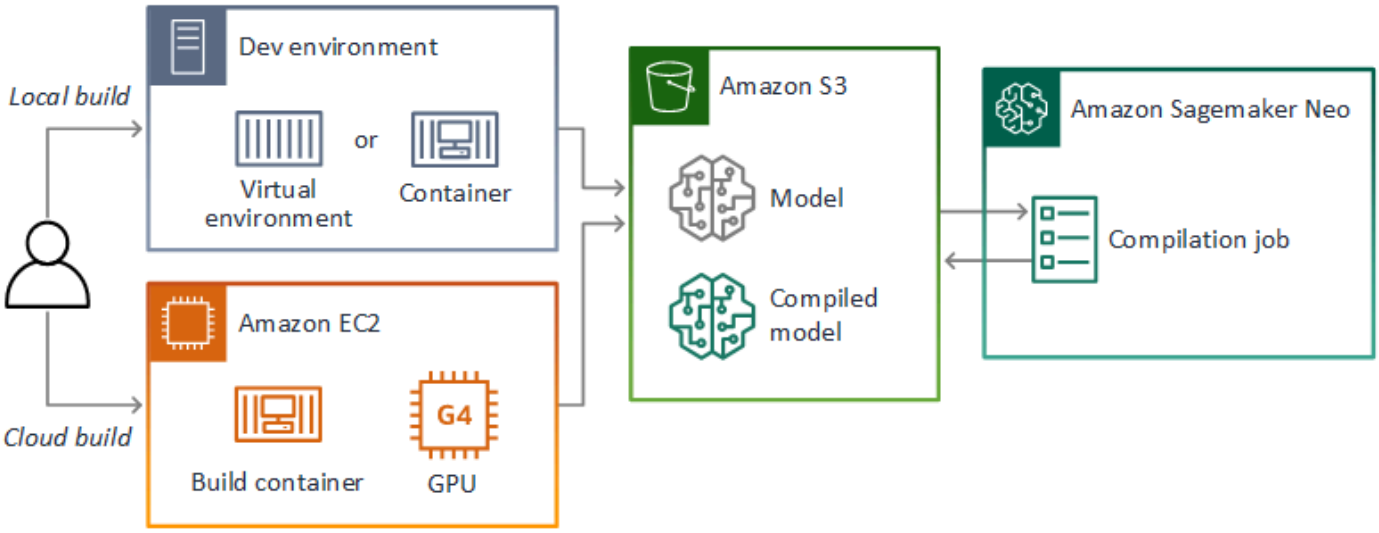
Criação de um modelo personalizado

Você pode usar modelos que você cria no PyTorch Apache MXNet e TensorFlow nos aplicativos do AWS Panorama. Como alternativa à criação e treinamento de modelos em SageMaker IA, você pode usar um modelo treinado ou criar e treinar seu próprio modelo com uma estrutura compatível e exportá-lo em um ambiente local ou na Amazon EC2.

> **Note**
>
> Para obter detalhes sobre as versões da estrutura e os formatos de arquivo suportados pelo SageMaker AI Neo, consulte Estruturas suportadas no Amazon SageMaker AI Developer Guide.

O repositório deste guia fornece um aplicativo de amostra que demonstra esse fluxo de trabalho para um modelo Keras em formato. TensorFlow `SavedModel` Ele usa TensorFlow 2 e pode ser executado localmente em um ambiente virtual ou em um contêiner Docker. O aplicativo de amostra também inclui modelos e scripts para criar o modelo em uma EC2 instância da Amazon.

- Aplicação de exemplo de modelo personalizado



O AWS Panorama usa o SageMaker AI Neo para compilar modelos para uso no AWS Panorama Appliance. Para cada estrutura, use o formato compatível com o SageMaker AI Neo e empacote o modelo em um `.tar.gz` arquivo.

Para obter mais informações, consulte Compilar e implantar modelos com o Neo no Amazon SageMaker AI Developer Guide.

Empacotamento de um modelo

Um pacote de modelos abrange um descritor, uma configuração de pacote e um arquivo de modelos. Assim como em um pacote de imagem da aplicação, a configuração do pacote informa ao serviço AWS Panorama onde o modelo e o descritor estão armazenados no 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 somente a versão principal e secundária da versão da estrutura. Para obter uma lista das versões compatíveis PyTorch, do Apache MXNet e TensorFlow das versões, consulte Estruturas suportadas.

Para importar um modelo, use o comando `import-raw-model` da CLI da aplicação do AWS Panorama. Se você fizer alguma alteração no modelo ou em seu descritor, será necessário executar novamente esse comando para atualizar os ativos da aplicação. Para obter mais informações, consulte Alteração do modelo de visão computacional.

Para o esquema JSON do arquivo descritor, consulte assetDescriptor.schema.json.

Modelos de treinamento

Ao treinar um modelo, use imagens do ambiente de destino ou de um ambiente de teste que se assemelhe muito ao ambiente de destino. Considere os seguintes fatores que podem afetar o desempenho do modelo:

- Iluminação: a quantidade de luz refletida por um objeto determina a quantidade de detalhes que o modelo precisa analisar. Um modelo treinado com imagens de objetos bem iluminados pode não funcionar bem em um ambiente com pouca luz ou baixa retroiluminação.
- Resolução: o tamanho da entrada de um modelo geralmente é fixado em uma resolução entre 224 e 512 pixels de largura em uma proporção quadrada. Antes de passar um quadro de vídeo para o modelo, você pode reduzi-lo ou cortá-lo para que ela caiba no tamanho necessário.
- Distorção da imagem: a distância focal e o formato da lente da câmera podem fazer com que as imagens exibam distorção longe do centro do quadro. A posição de uma câmera também determina quais atributos de um objeto são visíveis. Por exemplo, uma câmera suspensa com lente grande angular mostrará a parte superior de um objeto quando ele estiver no centro do quadro e uma visão distorcida da lateral do objeto à medida que ele se afasta do centro.

Para resolver esses problemas, você pode pré-processar imagens antes de enviá-las ao modelo e treinar o modelo com uma variedade maior de imagens que refletem variações em ambientes do mundo real. Se um modelo precisar operar em situações de iluminação e com várias câmeras, você precisará de mais dados para o treinamento. Além de coletar mais imagens, você pode obter mais dados de treinamento criando variações de suas imagens existentes que estão distorcidas ou têm iluminação diferente.

Construir uma imagem de aplicação

O AWS Panorama Appliance executa aplicações como sistemas de arquivos de contêiner exportados a partir de uma imagem que você cria. Você especifica as dependências e os recursos da sua aplicação em um Dockerfile que usa a imagem base da aplicação do AWS Panorama como ponto de partida.

Para criar uma imagem da aplicação, você usa o Docker e a CLI da aplicação do AWS Panorama. O exemplo a seguir da aplicação de exemplo deste guia demonstra esses 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
```

As seguintes instruções do Dockerfile são usadas.

- `FROM`: carrega a imagem base da aplicação (`public.ecr.aws/panorama/panorama-application`).
- `WORKDIR`: define o diretório de trabalho na imagem. `/panorama` é usado para o código da aplicação e arquivos relacionados. Essa configuração só persiste durante a compilação e não afeta o diretório de trabalho da sua aplicação em runtime (`/`).
- `COPY`: copia arquivos de um caminho local para um caminho na imagem. `COPY . .` copia os arquivos no diretório atual (o diretório do pacote) para o diretório de trabalho na imagem. Por exemplo, o código da aplicação é copiado de `packages/123456789012-SAMPLE_CODE-1.0/application.py` para `/panorama/application.py`.
- `RUN`: executa comandos de shell na imagem durante a compilação. Uma única operação `RUN` pode executar vários comandos em sequência usando `&&` entre comandos. Este exemplo atualiza o gerenciador de pacotes `pip` e, em seguida, instala as bibliotecas listadas em `requirements.txt`.

Você pode usar outras instruções, como `ADD` e `ARG`, que são úteis no momento da compilação. Instruções que adicionam informações de runtime ao contêiner, como `ENV`, não funcionam com o

AWS Panorama. O AWS Panorama não executa um contêiner a partir da imagem. Ele só usa a imagem para exportar um sistema de arquivos, que é transferido para o dispositivo.

Especificação de dependências

`requirements.txt` é um arquivo de requisitos do Python que especifica as bibliotecas usadas pela aplicação. A aplicação de exemplo usa Open CV e o AWS SDK para Python (Boto3).

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

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

O comando `pip install` no Dockerfile instala essas bibliotecas no diretório `dist-packages` Python abaixo de `/usr/local/lib`, para que elas possam ser importadas pelo código da sua aplicação.

Armazenamento local

O AWS Panorama reserva o diretório `/opt/aws/panorama/storage` para armazenamento de aplicações. Sua aplicação pode criar e modificar arquivos nesse caminho. Os arquivos criados no diretório de armazenamento persistem nas reinicializações. Outros locais de arquivos temporários são apagados na inicialização.

Criação de ativos de imagem

Quando você cria uma imagem para seu pacote de aplicações com a CLI da aplicação do AWS Panorama, a CLI executa `docker build` no diretório do pacote. Isso cria uma imagem da aplicação que contém o código da aplicação. Em seguida, a CLI cria um contêiner, exporta seu sistema de arquivos, o compacta e o armazena na pasta `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
```

O bloco JSON na saída é uma definição de ativo que a CLI adiciona à configuração do pacote (`package.json`) e registra no serviço AWS Panorama. A CLI também copia o arquivo descritor, que especifica o caminho para o script da aplicação (o ponto de entrada da aplicação).

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

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

Na pasta de ativos, o descritor e a imagem da aplicação são nomeados de acordo com sua soma de verificação SHA-256. Esse nome é usado como um identificador exclusivo para o ativo quando ele é armazenado no Amazon S3.

Chamada de serviços da AWS a partir do código da sua aplicação

Você pode usar o AWS SDK for Python (Boto) para chamar os serviços da AWS a partir do código do seu aplicativo. Por exemplo, se seu modelo detectar algo fora do comum, você pode publicar métricas na Amazon CloudWatch, enviar uma notificação com o Amazon SNS, salvar uma imagem no Amazon S3 ou invocar uma função Lambda para processamento adicional. A maioria dos serviços da AWS tem uma API pública que você pode usar com o SDK da AWS.

O dispositivo não tem permissão para acessar nenhum serviço da AWS por padrão. Para conceder permissão, crie uma função para a aplicação e atribua-a à instância da aplicação durante a implantação.

Seções

- Usar o Amazon S3
- Usando o tópico AWS IoT MQTT

Usar o Amazon S3

É possível usar o Amazon S3 para armazenar resultados de processamento e outros dados da aplicação.

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

Usando o tópico AWS IoT MQTT

É possível usar o SDK para Python (Boto3) para enviar mensagens para um tópico MQTT no AWS IoT. No exemplo a seguir, a aplicação publica um tópico com o nome da coisa do dispositivo, que você pode encontrar no AWS IoT console.

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

Escolha um nome que indique o ID do dispositivo ou outro identificador de sua escolha. Para publicar mensagens, a aplicação precisa de permissão para chamar `iot:Publish`.

Para monitorar uma fila MQTT

1. Abara a página Teste de console AWS IoT.
2. Em Tópico de assinatura, insira o nome do tópico. Por exemplo, `panorama/panorama_my-appliance_Thing_a01e373b`.
3. Escolha Assinar um tópico.

O SDK para aplicações do AWS Panorama

O SDK para aplicações do AWS Panorama é uma biblioteca Python para o desenvolvimento de aplicações do AWS Panorama. No código da sua aplicação, você usa o SDK para aplicações do AWS Panorama para carregar um modelo de visão computacional, executar inferência e enviar vídeo para um monitor.

> **Note**
>
> Para garantir que você tenha acesso às funcionalidades mais recentes do SDK para aplicações do AWS Panorama, atualize o software do dispositivo.

Para obter detalhes sobre as classes que o SDK para aplicações define e seus métodos, consulte a Referência do SDK para aplicações.

Seções

- Adição de texto e caixas à saída de vídeo

Adição de texto e caixas à saída de vídeo

Com o SDK do AWS Panorama, você pode enviar um stream de vídeo para um monitor. O vídeo pode incluir texto e caixas que mostram a saída do modelo, o estado atual da aplicação ou outros dados.

Cada objeto na matriz `video_in` é uma imagem de um stream de câmera conectado ao dispositivo. O tipo desse objeto é `panoramasdk.media`. Ele tem métodos para adicionar texto e caixas retangulares à imagem, que você pode então atribuir à matriz `video_out`.

No exemplo a seguir, a aplicação de exemplo adiciona um rótulo para cada um dos resultados. Cada resultado é posicionado na mesma posição à esquerda, mas em alturas diferentes.

```
        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 adicionar uma caixa à imagem de saída, use `add_rect`. Esse método usa 4 valores entre 0 e 1, indicando a posição dos cantos superior esquerdo e inferior direito da caixa.

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

Execução de vários threads

Você pode executar a lógica da aplicação em um thread de processamento e usar outros thread para outros processos em segundo plano. Por exemplo, você pode criar um encadeamento que veicule tráfego HTTP para depuração ou um encadeamento que monitore os resultados da inferência e envie dados para. AWS

Para executar vários threads, use o módulo de thread da biblioteca padrão do Python para criar um thread para cada processo. O exemplo a seguir mostra o loop principal da aplicação de exemplo do servidor de depuração, que cria um objeto de aplicação e o usa para executar três threads.

Example packages/123456789012-DEBUG_SERVER-1.0/application.py: loop 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.')
```

Quando todos os threads são encerrados, a aplicação se reinicia sozinha. O loop `run_cv` processa imagens de streams de câmera. Se ele receber um sinal para parar, ele desliga o processo do

depurador, que executa um servidor HTTP e não consegue se desligar sozinho. Cada thread deve lidar com seus próprios erros. Se um erro não for detectado e registrado em log, o thread será encerrado silenciosamente.

Example packages/123456789012-DEBUG_SERVER-1.0/application.py: Loop de processamento

```
    # 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")
```

Os threads se comunicam por meio do objeto `self` da aplicação. Para reiniciar o loop de processamento da aplicação, o thread do depurador chama o método `stop`. Esse método define um atributo `terminate` que sinaliza que os outros threads sejam encerrados.

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)
```

Fornecimento de tráfego de entrada

Você pode monitorar ou depurar aplicações localmente executando um servidor HTTP junto com o código da aplicação. Para fornecer tráfego externo, mapeie as portas no AWS Panorama Appliance para portas no contêiner do sua aplicação.

> **Important**
>
> Por padrão, o AWS Panorama Appliance não aceita tráfego de entrada em nenhuma porta. A abertura de portas no dispositivo tem um risco de segurança implícito. Ao usar esse atributo, você deve tomar medidas adicionais para proteger seu dispositivo contra tráfego externo e proteger as comunicações entre clientes autorizados e o dispositivo.
> O código de exemplo incluído neste guia serve para fins de demonstração e não implementa autenticação, autorização ou criptografia.

Você pode abrir portas na faixa de 8.000 a 9.000 no dispositivo. Essas portas, quando abertas, podem receber tráfego de qualquer cliente roteável. Ao implantar sua aplicação, você especifica quais portas abrir e mapeia as portas do dispositivo para as portas do contêiner da aplicação. O software do dispositivo encaminha o tráfego para o contêiner e envia as respostas de volta ao solicitante. As solicitações são recebidas na porta do dispositivo que você especificar, e as respostas são enviadas em uma porta efêmera aleatória.

Configuração de portas de entrada

Você especifica mapeamentos de portas em três locais na configuração da sua aplicação. No pacote de código `package.json`, você especifica a porta que o nó de código escuta em um bloco `network`. O exemplo a seguir declara que o nó escuta na porta 80.

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

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

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

No manifesto da aplicação, você declara uma regra de roteamento que mapeia uma porta no dispositivo para uma porta no contêiner de código da aplicação. O exemplo a seguir adiciona uma regra que mapeia a porta 8080 no dispositivo para a porta 80 no contêiner `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."
            }
        }
    ]
```

Ao implantar o aplicativo, você especifica as mesmas regras no console do AWS Panorama ou com um documento de substituição passado para a CreateApplicationInstanceAPI. Você deve fornecer essa configuração no momento da implantação para confirmar que deseja abrir portas no 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"
    }
}
```

Se a porta do dispositivo especificada no manifesto da aplicação estiver sendo usada por outra aplicação, você poderá usar o documento de substituição para escolher uma porta diferente.

Fornecimento de tráfego

Com as portas abertas no contêiner, você pode abrir um soquete ou executar um servidor para processar as solicitações recebidas. O exemplo `debug-server` mostra uma implementação básica de um servidor HTTP em execução junto com o código da aplicação de visão computacional.

> **Important**
>
> A implementação de exemplo não é segura para uso em produção. Para evitar tornar seu dispositivo vulnerável a ataques, você deve implementar controles de segurança apropriados em seu código e na configuração de rede.

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.')
```

O servidor aceita solicitações GET no caminho `/status` para recuperar algumas informações sobre a aplicação. Ele também aceita uma solicitação POST para `/restart` para reiniciar a aplicação.

Para demonstrar essa funcionalidade, a aplicação de exemplo executa um cliente HTTP em um thread separado. O cliente chama o caminho `/status` pela rede local logo após a inicialização e reinicia a aplicação alguns minutos depois.

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")
```

O loop principal gerencia os threads e reinicia a aplicação quando eles saem.

Example packages/123456789012-DEBUG_SERVER-1.0/application.py: loop 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 implantar o aplicativo de amostra, consulte as instruções no GitHub repositório deste guia.

Uso da GPU

Você pode acessar o processador gráfico (GPU) no AWS Panorama Appliance para usar bibliotecas aceleradas por GPU ou executar modelos de machine learning no código da sua aplicação. Para ativar o acesso à GPU, adicione o acesso à GPU como requisito na configuração do pacote depois de criar o contêiner de código da aplicação.

> **Important**
>
> Se você habilitar o acesso à GPU, não poderá executar nós de modelo em nenhuma aplicação no dispositivo. Por motivos de segurança, o acesso à GPU é restrito quando o equipamento executa um modelo compilado com o SageMaker AI Neo. Com o acesso à GPU, você deve executar seus modelos nos nós de código da aplicação, e todas as aplicações no dispositivo compartilham o acesso à GPU.

Para ativar o acesso à GPU para sua aplicação, atualize a configuração do pacote depois de criar o pacote com a CLI da aplicação do AWS Panorama. O exemplo a seguir mostra o bloco `requirements` que adiciona acesso à GPU ao nó do código da aplicação.

Example package.json com bloco 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": [
    ...
```

Atualize a configuração do pacote entre as etapas de compilação e empacotamento em seu fluxo de trabalho de desenvolvimento.

Para implantar uma aplicação com acesso à GPU

1. Para criar o contêiner da aplicação, use o comando `build-container`.

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

2. Adicione o bloco `requirements` à configuração do pacote.
3. Para carregar o ativo do contêiner e a configuração do pacote, use o comando `package-application`.

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

4. Implante o aplicativo .

Para exemplos de aplicativos que usam acesso à GPU, visite o aws-panorama-samples GitHub repositório.

Configurar um ambiente de desenvolvimento no Windows

Para criar uma aplicação do AWS Panorama, você usa o Docker, ferramentas de linha de comando e Python. No Windows, você pode configurar um ambiente de desenvolvimento usando o Docker Desktop com o Subsistema do Windows para Linux e Ubuntu. Este tutorial mostra o processo de configuração de um ambiente de desenvolvimento que foi testado com ferramentas e aplicações de exemplo do AWS Panorama.

Seções

Pré-requisitos

Para seguir este tutorial, você precisa de uma versão do Windows que ofereça suporte ao Subsistema do Windows para Linux 2 (WSL 2).

- Windows 10 versão 1903 e superior (compilação 18362 e superior) ou Windows 11
- Atributos do Windows
  - Subsistema Windows para Linux
  - Hyper-V
  - Plataforma de máquina virtual

Este tutorial foi desenvolvido com as seguintes versões de software.

- Ubuntu 20.04
- Python 3.8.5
- Docker 20.10.8

Instalação do WSL 2 e do Ubuntu

Se você tiver o Windows 10 versão 2004 e superior (compilação 19041 e superior), poderá instalar o WSL 2 e o Ubuntu 20.04 com o comando a seguir. PowerShell

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

Para versões mais antigas do Windows, siga as instruções na documentação do WSL 2: Etapas de instalação manual para versões mais antigas

Instalar o Docker

Para instalar o Docker Desktop, baixe e execute o pacote do instalador em hub.docker.com. Se você encontrar problemas, siga as instruções no site do Docker: Docker Desktop WSL 2 backend.

Execute o Docker Desktop e siga o tutorial de primeira execução para criar um contêiner de exemplo.

> **Note**
>
> O Docker Desktop só ativa o Docker na distribuição padrão. Se você tiver outras distribuições Linux instaladas antes de executar este tutorial, habilite o Docker na distribuição Ubuntu recém-instalada no menu de configurações do Docker Desktop em Resources, WSL integration.

Configuração do Ubuntu

Agora você pode executar comandos do Docker na sua máquina virtual Ubuntu. Para abrir um terminal de linha de comando, execute a distribuição no menu Iniciar. Na primeira execução, configure um nome de usuário e uma senha que podem ser usados para executar comandos do administrador.

Para concluir a configuração do seu ambiente de desenvolvimento, atualize o software da máquina virtual e instale as ferramentas.

Para configurar a máquina virtual

1. Atualize o software que vem com o Ubuntu.

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

2. Instale ferramentas de desenvolvimento com apt.

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

3. Instale as bibliotecas Python com o pip.

```
$ pip3 install awscli panoramacli
```

4. Abra um novo terminal e, em seguida, execute `aws configure` para configurar a AWS CLI.

```
$ aws configure
```

   Se não tiver chaves de acesso, você poderá criá-las no console do IAM.

Por fim, baixe e importe a aplicação de exemplo.

Para obter a aplicação de exemplo

1. Faça o download e extraia a aplicação de exemplo.

```
$ 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. Execute os scripts incluídos para testar a compilação, criar o contêiner da aplicação e fazer upload de pacotes para o 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
```

A CLI da aplicação do AWS Panorama carrega pacotes e os registra no serviço AWS Panorama. Agora você pode implantar a aplicação de exemplo com o console do AWS Panorama.

Próximas etapas

Para explorar e editar os arquivos do projeto, você pode usar o Explorador de arquivos ou um ambiente de desenvolvimento integrado (IDE) que suporte WSL.

Para acessar o sistema de arquivos da máquina virtual, abra o Explorador de arquivos e digite `\\wsl$` na barra de navegação. Esse diretório contém um link para o sistema de arquivos da máquina virtual (`Ubuntu-20.04`) e sistemas de arquivos para os dados do Docker. Em `Ubuntu-20.04`, seu diretório de usuários está em home*username*.

> **Note**
>
> Para acessar arquivos em sua instalação do Windows a partir do Ubuntu, navegue até o diretório `/mnt/c`. Por exemplo, você pode listar arquivos em seu diretório de downloads executando `ls /mnt/c/Users/`*windows-username*`/Downloads`.

Com o Visual Studio Code, você pode editar o código da aplicação em seu ambiente de desenvolvimento e executar comandos com um terminal integrado. Para instalar o Visual Studio Code, visite code.visualstudio.com. Após a instalação, adicione a extensão Remote WSL.

O Windows Terminal é uma alternativa ao terminal padrão do Ubuntu no qual você está executando comandos. Ele suporta várias guias e pode ser executado PowerShell, prompt de comando e terminais para qualquer outra variedade de Linux que você instalar. Ele suporta copiar e colar com Ctrl+C eCtrl+V, clicável URLs e outras melhorias úteis. Para instalar o Windows Terminal, visite microsoft.com.

A API do AWS Panorama

Você pode usar a API pública do serviço AWS Panorama para automatizar fluxos de trabalho de gerenciamento de dispositivos e aplicações. Com o AWS Command Line Interface ou o AWS SDK, você pode desenvolver scripts ou aplicativos que gerenciam recursos e implantações. O GitHub repositório deste guia inclui scripts que você pode usar como ponto de partida para seu próprio código.

- aws-panorama-developer-guide/scripts de utilitários

Seções

Automatização do registro de dispositivos

Para provisionar um dispositivo, use a ProvisionDeviceAPI. A resposta inclui um arquivo ZIP com a configuração e as credenciais temporárias do dispositivo. Decodifique o arquivo e salve-o em um arquivamento com o prefixo `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}"
```

As credenciais no arquivo de configuração expiram após 5 minutos. Transfira o arquivo para o seu dispositivo com a unidade USB incluída.

Para registrar uma câmera, use a CreateNodeFromTemplateJobAPI. Essa API usa um mapa dos parâmetros do modelo para o nome de usuário, a senha e o URL da câmera. Você pode formatar esse mapa como um documento JSON usando a manipulação de strings do 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, você pode carregar a configuração JSON de um arquivo.

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

Gerenciamento de dispositivos com a API do AWS Panorama

Você pode automatizar tarefas de gerenciamento de dispositivos com a API do AWS Panorama.

Exibição de dispositivos

Para obter uma lista de dispositivos com dispositivo IDs, use a ListDevicesAPI.

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

Para obter mais detalhes sobre um dispositivo, use a 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
}
```

Atualizar o software do dispositivo

Se a versão `LatestSoftware` for mais recente que a versão `CurrentSoftware`, você poderá atualizar o dispositivo. Use a CreateJobForDevicesAPI para criar um trabalho de atualização 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"
        }
    ]
}
```

Em um script, você pode preencher o campo da versão da imagem no arquivo de configuração do trabalho com a manipulação de string do 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
}
```

O dispositivo baixa a versão do software especificada e se atualiza sozinho. Veja o progresso da atualização com a 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 obter uma lista de todos os trabalhos em execução, use ListDevicesJobso.

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

Para ver um exemplo de script que verifica e aplica atualizações, consulte check-updates.sh no GitHub repositório deste guia.

Reinicialização de dispositivos

Para reinicializar um dispositivo, use a CreateJobForDevicesAPI.

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

```
    ]
}
```

Em um script, você pode obter uma lista de dispositivos e escolher um para reinicializar interativamente.

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"
        }
    ]
}
```

Automatização da implantação da aplicação

Para implantar um aplicativo, você usa o AWS Panorama Application CLI e. AWS Command Line Interface Depois de criar o contêiner da aplicação, faça o upload dele e de outros ativos para um ponto de acesso Amazon S3. Em seguida, você implanta o aplicativo com a CreateApplicationInstanceAPI.

Para obter mais contexto e instruções sobre como usar os scripts mostrados, siga as instruções no README da aplicação de exemplo.

Seções

Crie o contêiner

Para criar o contêiner da aplicação, use o comando `build-container`. Esse comando cria um contêiner do Docker e o salva como um sistema de arquivos compactado na pasta `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
```

Você também pode usar o preenchimento da linha de comando para preencher o argumento do caminho digitando parte do caminho e pressionando TAB.

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

Upload do contêiner e registro dos nós

Para carregar a aplicação, use o comando `package-application`. Esse comando carrega ativos da pasta `assets` para um ponto de acesso Amazon S3 que o AWS Panorama gerencia.

Example 4-package-app.sh

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

A CLI da aplicação do AWS Panorama carrega ativos de contêineres e descritores referenciados pela configuração do pacote (`package.json`) em cada pacote e registra os pacotes como nós no AWS Panorama. Em seguida, referencie esses nós no seu manifesto (`graph.json`) da aplicação para implantar a aplicação.

Implantar a aplicação

Para implantar o aplicativo, você usa a CreateApplicationInstanceAPI. Essa ação usa os seguintes parâmetros, entre outros.

- `ManifestPayload`: o manifesto da aplicação (`graph.json`) que define os nós, pacotes, bordas e parâmetros da aplicação.
- `ManifestOverridesPayload`: um segundo manifesto que substitui os parâmetros do primeiro. O manifesto da aplicação pode ser considerado um recurso estático na origem da aplicação, em que o manifesto de substituição fornece configurações de tempo de implantação que personalizam a implantação.
- `DefaultRuntimeContextDevice`: o dispositivo de destino.
- `RuntimeRoleArn`: o ARN de um perfil do IAM que a aplicação usa para acessar os serviços e recursos da AWS.
- `ApplicationInstanceIdToReplace`: o ID de uma instância de aplicação existente a ser removida do dispositivo.

O manifesto e as cargas de substituição são documentos JSON que devem ser fornecidos como um valor de string aninhado dentro de outro documento. Para fazer isso, o script carrega os manifestos de um arquivo como uma string e usa a ferramenta jq para construir o documento aninhado.

Example 5-deploy.sh: componha manifestos

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

O script de implantação usa a ListDevicesAPI para obter uma lista de dispositivos registrados na região atual e salva a escolha do usuário em um arquivo local para implantações subsequentes.

Example 5-deploy.sh: encontre um 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)
```

O perfil da aplicação é criado por outro script (1-create-role.sh). O script de implantação obtém o ARN dessa função de. AWS CloudFormation Se a aplicação já estiver implantada no dispositivo, o script obterá o ID dessa instância da aplicação a partir de um arquivo local.

Example 5-deploy.sh: ARN do perfil e argumentos de substituição

```
# 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 fim, o script reúne todas as peças para criar uma instância da aplicação e implantar a aplicação no dispositivo. O serviço responde com um ID de instância que o script armazena para uso posterior.

Example 5-deploy.sh: implante a aplicação

```
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
```

Monitore a implantação

Para monitorar uma implantação, use a ListApplicationInstancesAPI. O script de monitor obtém o ID do dispositivo e o ID da instância da aplicação dos arquivos no diretório da aplicação e os usa para criar um comando da CLI. Em seguida, ele chama em um loop.

Example 6-monitor-deployment.sh

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

Quando a implantação for concluída, você poderá visualizar os registros chamando a API Amazon CloudWatch Logs. O script de visualização de registros usa a `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
```

Gerenciamento de aplicações com a API AWS Panorama

Você pode monitorar e gerenciar aplicações com a API do AWS Panorama.

Visualizar aplicações

Para obter uma lista dos aplicativos em execução em um dispositivo, use a 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 obter mais detalhes sobre os nós de uma instância de aplicativo, use a 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"
        }
    ]
}
```

Gerenciamento de streams de câmera

Você pode pausar e retomar os nós de transmissão da câmera com a [SignalApplicationInstanceNodeInstances](#)API.

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

Em um script, você pode obter uma lista de nós e escolher um para pausar ou retomar interativamente.

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"
}
```

Ao pausar e retomar os nós da câmera, você pode percorrer um número maior de streams de câmera do que os que podem ser processados simultaneamente. Para fazer isso, mapeie vários streams de câmera para o mesmo nó de entrada em seu manifesto de substituição.

No exemplo a seguir, o manifesto de substituição mapeia dois streams de câmera ,`warehouse-floor` e `entrance-north`, para o mesmo nó de entrada (`camera_node`). O stream `warehouse-floor` fica ativo quando a aplicação é iniciada, e o nó `entrance-north` espera que um sinal seja ativado.

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 obter detalhes sobre a implantação com a API, consulte Automatização da implantação da aplicação.

Usar endpoints da VPC

Se você trabalha em uma VPC sem acesso à Internet, pode criar um endpoint da VPC para uso com o AWS Panorama. Um endpoint da VPC permite que clientes em execução em uma sub-rede privada se conectem a um serviço da AWS sem conexão com a Internet.

Para obter detalhes sobre portas e endpoints usados pelo AWS Panorama Appliance, consulte ???.

Seções

- Criar um endpoint da VPC
- Conexão de um dispositivo a uma sub-rede privada
- AWS CloudFormation Modelos de amostra

Criar um endpoint da VPC

Para estabelecer uma conexão privada entre sua VPC e o AWS Panorama, crie um endpoint da VPC. Não é necessário um endpoint da VPC para usar o AWS Panorama. Você só precisa criar um endpoint da VPC se trabalhar em uma VPC sem acesso à Internet. Quando a CLI ou o SDK da AWS tenta se conectar ao AWS Panorama, o tráfego é roteado pelo endpoint da VPC.

Crie um endpoint da VPC para o AWS Panorama usando as seguintes configurações:

- Nome de serviço : **`com.amazonaws.`*`us-west-2`*`.panorama`**
- Tipo: interface

Um endpoint da VPC usa o nome DNS do serviço para obter tráfego de clientes do SDK da AWS sem nenhuma configuração adicional. Para obter mais informações sobre endpoints da VPC, consulte Endpoints da VPC de interface no Guia do usuário do Amazon VPC.

Conexão de um dispositivo a uma sub-rede privada

O AWS Panorama Appliance pode se conectar AWS por meio de uma conexão VPN privada com AWS Site-to-Site VPN ou AWS Direct Connect. Com esses serviços, você pode criar uma sub-rede privada que se estende até seu datacenter. O dispositivo se conecta à sub-rede privada e acessa os serviços AWS por meio de VPC endpoints.

Site-to-Site VPN e AWS Direct Connect são serviços para conectar seu data center à Amazon VPC com segurança. Com a Site-to-Site VPN, você pode usar dispositivos de rede disponíveis comercialmente para se conectar. AWS Direct Connect usa um AWS dispositivo para se conectar.

- Site-to-Site VPN — O que é AWS Site-to-Site VPN?
- AWS Direct Connect: O que é AWS Direct Connect?

Depois de conectar sua rede local a uma sub-rede privada em uma VPC, crie endpoints da VPC para os seguintes serviços.

- Amazon Simple Storage Service: AWS PrivateLink para Amazon S3
- AWS IoT Core: uso do AWS IoT Core com endpoints da VPC da interface (plano de dados e provedor de credenciais)
- Amazon Elastic Container Registry: endpoints da VPC da interface do Amazon Elastic Container Registry
- Amazon CloudWatch — Usando CloudWatch com interface VPC endpoints
- Amazon CloudWatch Logs — Usando CloudWatch registros com endpoints de interface VPC

O dispositivo não precisa de conectividade com o serviço AWS Panorama. Ele se comunica com o AWS Panorama por meio de um canal de mensagens em AWS IoT.

Além dos endpoints VPC, o Amazon S3 exige AWS IoT o uso de zonas hospedadas privadas do Amazon Route 53. A zona hospedada privada encaminha o tráfego de subdomínios, incluindo subdomínios para pontos de acesso Amazon S3 e tópicos do MQTT, para o endpoint da VPC correto. Para obter mais informações sobre zonas hospedadas privadas, consulte Trabalhar com zonas hospedadas privadas no Guia do desenvolvedor do Amazon Route 53.

Para ver um exemplo de configuração de VPC com endpoints da VPC e zonas hospedadas privadas, consulte AWS CloudFormation Modelos de amostra.

AWS CloudFormation Modelos de amostra

O GitHub repositório deste guia fornece AWS CloudFormation modelos que você pode usar para criar recursos para uso com o AWS Panorama. Os modelos criam uma VPC com duas sub-redes privadas, uma sub-rede pública e um endpoint da VPC. Você pode usar as sub-redes privadas

na VPC para hospedar recursos isolados da Internet. Os recursos na sub-rede pública podem se comunicar com os recursos privados, mas os recursos privados não podem ser acessados pela Internet.

Example vpc-endpoint.yml: sub-redes 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
  ...
```

O modelo `vpc-endpoint.yml` mostra como criar um endpoint da VPC para o AWS Panorama. Você pode usar esse endpoint para gerenciar recursos do AWS Panorama com o AWS SDK ou. AWS CLI

Example vpc-endpoint.yml: endpoint da 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:
            - "*"
```

O `PolicyDocument` é uma política de permissões baseada em recursos que define as chamadas de API que podem ser feitas com o endpoint. Você pode modificar a política para restringir as ações e os recursos que podem ser acessados por meio do endpoint. Para obter mais informações, consulte Controlar o acesso a serviços com endpoints da VPC no Guia do Usuário do Amazon VPC.

O modelo `vpc-appliance.yml` mostra como criar endpoints da VPC e zonas hospedadas privadas para serviços usados pelo AWS Panorama Appliance.

Example vpc-appliance.yml: endpoint do ponto de acesso Amazon S3 com zona hospedada 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 ] ] ]
```

Os modelos de exemplo demonstram a criação de recursos da Amazon VPC e do Route 53 com uma VPC de exemplo. Você pode adaptá-los ao seu caso de uso removendo os recursos da VPC e substituindo as referências à sub-rede, ao grupo de segurança e à VPC IDs pelos dos seus recursos. IDs

Exemplos de aplicações, scripts e modelos

O GitHub repositório deste guia fornece exemplos de aplicativos, scripts e modelos para AWS Panorama dispositivos. Use esses exemplos para aprender as melhores práticas e automatizar os fluxos de trabalho de desenvolvimento.

Seções

Aplicações de exemplo

Exemplos de aplicativos demonstram o uso de AWS Panorama recursos e tarefas comuns de visão computacional. Essas aplicações de exemplo incluem scripts e modelos que automatizam a configuração e a implantação. Com uma configuração mínima, você pode implantar e atualizar aplicações via linha de comando.

- aws-panorama-sample— Visão computacional básica com um modelo de classificação. Use o AWS SDK for Python (Boto) para fazer upload de métricas CloudWatch, instrumentar métodos de pré-processamento e inferência e configurar o registro.
- debug-server: abra as portas de entrada no dispositivo e encaminhe o tráfego para um contêiner de código da aplicação. Use multithreading para executar o código da aplicação, um servidor HTTP e um cliente HTTP simultaneamente.
- modelo personalizado — Exporte modelos do código e compile com o SageMaker AI Neo para testar a compatibilidade com o Appliance. AWS Panorama Crie localmente em um desenvolvimento em Python, em um contêiner Docker ou em uma instância da Amazon. EC2 Exporte e compile todos os modelos de aplicativos integrados no Keras para uma versão específica ou em TensorFlow Python.

Para ver mais exemplos de aplicativos, visite também o aws-panorama-samplesrepositório.

Scripts de utilitários

Os scripts no `util-scripts` diretório gerenciam AWS Panorama recursos ou automatizam fluxos de trabalho de desenvolvimento.

- provision-device.sh: provisione um dispositivo.
- check-updates.sh: verifique e aplique as atualizações do software do dispositivo.
- reboot-device.sh: reinicie um dispositivo.
- register-camera.sh: registre uma câmera.
- deregister-camera.sh: exclui um nó de câmera.
- view-logs.sh: visualize os logs de uma instância da aplicação.
- pause-camera.sh: pause ou retome um stream da câmera.
- push.sh: crie, faça upload e implante uma aplicação.
- rename-package.sh: renomeie um pacote de nós. Atualiza os nomes dos diretórios, os arquivos de configuração e o manifesto da aplicação.
- samplify.sh: substitua o ID da sua conta por um exemplo de ID de conta e restaure as configurações de backup para remover a configuração local.
- update-model-config.sh — Adicione novamente o modelo ao aplicativo depois de atualizar o arquivo descritor.
- cleanup-patches.sh: cancele o registro de versões antigas do patch e exclua seus manifestos do Amazon S3.

Para obter detalhes de uso, consulte o README.

AWS CloudFormation modelos

Use os AWS CloudFormation modelos no `cloudformation-templates` diretório para criar recursos para AWS Panorama aplicativos.

- alarm-application.yml: crie um alarme que monitore erros na aplicação. Se a instância da aplicação gerar erros ou parar de funcionar por 5 minutos, o alarme enviará um e-mail de notificação.
- alarm-device.yml: crie um alarme que monitore a conectividade de um dispositivo. Se o dispositivo parar de enviar métricas por 5 minutos, o alarme enviará um e-mail de notificação.

- application-role.yml: crie um perfil da aplicação. A função inclui permissão para enviar métricas para CloudWatch. Adicione permissões à declaração de política para outras operações de API que sua aplicação usa.
- vpc-appliance.yml — Crie uma VPC com acesso privado ao serviço de sub-rede para o equipamento. AWS Panorama Para conectar o dispositivo a uma VPC, AWS Direct Connect use ou. AWS Site-to-Site VPN
- vpc-endpoint.yml — Crie uma VPC com acesso ao serviço de sub-rede privada. AWS Panorama Os recursos dentro da VPC podem se conectar AWS Panorama para monitorar e gerenciar AWS Panorama recursos sem se conectar à Internet.

O `create-stack.sh` script nesse diretório cria AWS CloudFormation pilhas. É preciso um número variável de argumentos. O primeiro argumento é o nome do modelo, e os argumentos restantes são substituições de parâmetros no modelo.

Por exemplo, o seguinte comando cria uma nova aplicação usando uma função do aplicativo.

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

Mais exemplos e ferramentas

O aws-panorama-samplesrepositório tem mais aplicativos de amostra e ferramentas úteis.

- Aplicações: aplicações de exemplo para várias arquiteturas de modelos e casos de uso.
- Validação do stream da câmera: valide os streams de câmera.
- PanoJupyter— Execute JupyterLab em um AWS Panorama dispositivo.
- Sideloading: atualize o código da aplicação sem criar ou implantar um contêiner de aplicação.

A AWS comunidade também desenvolveu ferramentas e orientações para AWS Panorama. Confira os seguintes projetos de código aberto em GitHub.

- cookiecutter-panorama — Um modelo Cookiecutter para aplicativos. AWS Panorama
- backpack: módulos Python para acessar detalhes do ambiente de runtime, perfis e opções adicionais de saída de vídeo.

Monitorando AWS Panorama recursos e aplicativos

Você pode monitorar AWS Panorama recursos no AWS Panorama console e com a Amazon CloudWatch. O AWS Panorama dispositivo se conecta à AWS nuvem pela Internet para relatar seu status e o status das câmeras conectadas. Enquanto está ligado, o equipamento também envia registros para o CloudWatch Logs em tempo real.

O dispositivo obtém permissão para usar AWS IoT, CloudWatch Logs e outros serviços da AWS a partir de uma função de serviço que você cria na primeira vez que usa o AWS Panorama console. Para obter mais informações, consulte Perfis de serviço e recursos entre serviços do AWS Panorama.

Para obter ajuda na solução de erros específicos, consulte Solução de problemas.

Tópicos

Monitoramento no console do AWS Panorama

Você pode usar o console do AWS Panorama para monitorar seu AWS Panorama Appliance e suas câmeras. O console é usado AWS IoT para monitorar o estado do equipamento.

Para monitorar seu dispositivo no console do AWS Panorama

1. Abra o console do AWS Panorama.
2. Abra a página Dispositivos do console do AWS Panorama.
3. Escolha um dispositivo.
4. Para ver o status de uma instância da aplicação, escolha-a na lista.
5. Para ver o status das interfaces de rede do dispositivo, escolha Configurações.

O status geral do dispositivo é exibido na parte superior da página. Se o status for Online, o equipamento estará conectado AWS e enviando atualizações de status regulares.

Visualização dos logs do AWS Panorama

O AWS Panorama relata eventos de aplicativos e sistemas para o Amazon CloudWatch Logs. Ao encontrar problemas, você pode usar os logs de eventos para ajudar a depurar sua aplicação do AWS Panorama ou solucionar problemas de configuração da aplicação.

Para ver registros em CloudWatch Registros

1. Abra a página Grupos de CloudWatch registros do console de registros.
2. Encontre logs de aplicações e dispositivos do AWS Panorama nos seguintes grupos:

   - Logs do dispositivo: `/aws/panorama/devices/device-id`
   - Logs da aplicação: `/aws/panorama/devices/device-id/applications/instance-id`

Ao reprovisionar um dispositivo após atualizar o software do sistema, você também pode visualizar os logs na unidade USB de provisionamento.

Seções

Visualizar logs do dispositivo

O AWS Panorama Appliance cria um grupo de logs para o dispositivo e um grupo para cada instância da aplicação que você implantar. Os logs do dispositivo contêm informações sobre o status da aplicação, atualizações de software e configuração do sistema.

Logs do dispositivo: **`/aws/panorama/devices/device-id`**

- `occ_log`: saída do processo do controlador. Esse processo coordena as implantações de aplicações e relata o status dos nós de cada instância da aplicação.
- `ota_log`— Saída do processo que coordena as atualizações de software over-the-air (OTA).

- `syslog`: saída do processo syslog do dispositivo, que captura as mensagens enviadas entre os processos.
- `kern_log`: eventos do kernel Linux do dispositivo.
- `logging_setup_logs`— Saída do processo que configura o agente CloudWatch Logs.
- `cloudwatch_agent_logs`— Saída do agente CloudWatch Logs.
- `shadow_log`: saída da sombra do dispositivo AWS IoT.

Visualizar logs da aplicação

O grupo de logs de uma instância de aplicação contém um fluxo de logs para cada nó, com o nome do nó.

Logs da aplicação: **`/aws/panorama/devices/`*`device-id`*`/applications/`*`instance-id`***

- Código: saída do código da sua aplicação e do SDK para aplicações do AWS Panorama. Agrega os logs da aplicação de `/opt/aws/panorama/logs`.
- Modelo: saída do processo que coordena as solicitações de inferência com um modelo.
- Stream: saída do processo que decodifica o vídeo de um stream de câmera.
- Monitor: saída do processo que renderiza a saída de vídeo para a porta HDMI.
- `mds`: logs do servidor de metadados do dispositivo.
- `console_output`: captura streams de saída padrão e de erros dos contêineres de código.

Se você não vê registros em CloudWatch Logs, confirme se você está na região correta da AWS. Se você estiver, pode haver um problema com a conexão do dispositivo com a AWS ou com as permissões na função do dispositivo AWS Identity and Access Management (IAM).

Configuração de logs da aplicação

Configure um logger em Python para gravar arquivos de log em `/opt/aws/panorama/logs`. O equipamento transmite registros desse local para CloudWatch Logs. Para evitar usar muito espaço em disco, use um tamanho máximo de arquivo de 10 MiB e uma contagem de backup de 1. O exemplo a seguir mostra um método que cria um logger.

Example application.py: configuração do logger

```
def get_logger(name=__name__,level=logging.INFO):
```

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

Inicialize o logger no escopo global e use-o em todo o código da aplicação.

Example application.py: inicializa o logger

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

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

Visualização de logs de provisionamento

Durante o provisionamento, o AWS Panorama Appliance copia os logs para a unidade USB que você usa para transferir o arquivo de configuração para o dispositivo. Use esses logs para solucionar problemas de provisionamento em dispositivos com a versão mais recente do software.

> **Important**
>
> Os logs de provisionamento estão disponíveis para dispositivos atualizados para a versão de software 4.3.23 ou mais recente.

Logs de aplicações

- `/panorama/occ.log`: logs do software controlador do AWS Panorama.
- `/panorama/ota_agent.log`— Registros do agente de over-the-air atualização do AWS Panorama.
- `/panorama/syslog.log`: logs do sistema Linux.
- `/panorama/kern.log`: logs de kernel do Linux.

Saída de logs de um dispositivo

Se os registros do dispositivo e do aplicativo não aparecerem nos CloudWatch Registros, você poderá usar uma unidade USB para obter uma imagem de registro criptografada do dispositivo. A equipe de serviço do AWS Panorama pode descriptografar os logs em seu nome e ajudar na depuração.

Pré-requisitos

Para seguir o procedimento, você precisará do seguinte hardware:

- Unidade USB — Uma unidade FAT32 de memória flash USB formatada com pelo menos 1 GB de armazenamento, para transferir os arquivos de log do AWS Panorama Appliance.

Para extrair logs do dispositivo

1. Prepare uma unidade USB com uma pasta `managed_logs` dentro de uma pasta `panorama`.

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

2. Conecte a unidade USB ao dispositivo.
3. Desligue o AWS Panorama Appliance.
4. Ligue o AWS Panorama Appliance.
5. O dispositivo copia os logs para o dispositivo. O LED de status pisca em azul enquanto isso está em andamento.
6. Os arquivos de log podem então ser encontrados dentro do diretório `managed_logs` com o formato `panorama_device_log_`*`v1_dd_hh_mm`*`.img`

Não é possível decifrar a imagem do log por conta própria. Entre em contato com o suporte ao cliente, com um gerente técnico de contas do AWS Panorama ou com um arquiteto de soluções para coordenar com a equipe de serviço.

Monitorando dispositivos e aplicativos com a Amazon CloudWatch

Quando um dispositivo está on-line, o AWS Panorama envia métricas para a Amazon CloudWatch. Você pode criar gráficos e painéis com essas métricas no CloudWatch console para monitorar a atividade do equipamento e definir alarmes que o notificam quando os dispositivos ficam off-line ou os aplicativos encontram erros.

Para visualizar métricas no CloudWatch console

1. Abra a página de métricas do console do AWS Panorama (namespace `PanoramaDeviceMetrics`).
2. Escolha um esquema de dimensão.
3. Escolha métricas para adicioná-las ao gráfico.
4. Para escolher uma estatística diferente e personalizar o gráfico, use as opções na guia Graphed metrics (Métricas no gráfico). Por padrão, os gráficos usam a estatística `Average` para todas as métricas.

> **Preços**
>
> CloudWatch tem um nível Always Free. Além do limite do nível gratuito, CloudWatch cobramos por métricas, painéis, alarmes, registros e insights. Para obter detalhes, consulte Definição de preço do CloudWatch .

Para obter mais informações sobre CloudWatch, consulte o Guia CloudWatch do usuário da Amazon.

Seções

- Uso de métricas de dispositivos
- Uso de métricas da aplicação
- Configurar alarmes

Uso de métricas de dispositivos

Quando um dispositivo está on-line, ele envia métricas para a Amazon CloudWatch. Você pode usar essas métricas para monitorar a atividade do dispositivo e acionar um alarme se os dispositivos ficarem off-line.

- `DeviceActive`: enviado periodicamente quando o dispositivo está ativo.

  Dimensões: `DeviceId` e `DeviceName`.

Visualize a métrica `DeviceActive` com a estatística `Average`.

Uso de métricas da aplicação

Quando um aplicativo encontra um erro, ele envia métricas para a Amazon. CloudWatch Você pode usar essas métricas para acionar um alarme se alguma aplicação parar de funcionar.

- `ApplicationErrors`: o número de erros de aplicações registrados.

  Dimensões: `ApplicationInstanceName` e `ApplicationInstanceId`.

Visualize as métricas da aplicação com a estatística `Sum`.

Configurar alarmes

Para receber notificações quando uma métrica ultrapassa um limite, crie um alarme. Por exemplo, é possível criar um alarme que envia uma notificação quando a soma da métrica `ApplicationErrors` fica em 1 por 20 minutos.

Para criar um alarme

1. Abra a página de alarmes CloudWatch do console Amazon.
2. Selecione Criar alarme.
3. Escolha Selecionar métrica e localize uma métrica para seu dispositivo, como `ApplicationErrors` para `applicationInstance-gk75xmplqbqtenlnmz4ehiu7xa`, `my-application`.
4. Siga as instruções para configurar uma condição, uma ação e um nome para o alarme.

Para obter instruções detalhadas, consulte Criar um CloudWatch alarme no Guia do CloudWatch usuário da Amazon.

Solução de problemas

Os tópicos a seguir fornecem dicas de solução de problemas para erros e problemas que você pode encontrar ao usar o AWS Panorama console, o equipamento ou o SDK. Se encontrar um problema que não esteja listado aqui, você poderá usar o botão Provide feedback desta página para relatá-lo.

Você pode encontrar registros do seu dispositivo no console do Amazon CloudWatch Logs. O dispositivo carrega logs do código da aplicação, do software do dispositivo e dos processos AWS IoT à medida que são gerados. Para obter mais informações, consulte Visualização dos logs do AWS Panorama.

Provisionamento

Problema: (macOS) Meu computador não reconhece a unidade USB incluída com um adaptador USB-C.

Isso pode ocorrer se você conectar a unidade USB a um adaptador USB-C que já esteja conectado ao seu computador. Tente desconectar o adaptador e reconectá-lo com a unidade USB já conectada.

Problema: o provisionamento falha quando eu uso minha própria unidade USB.

Problema: o provisionamento falha quando eu uso a porta USB 2.0 do dispositivo.

O AWS Panorama aparelho é compatível com dispositivos de memória flash USB entre 1 e 32 GB, mas nem todos são compatíveis. Alguns problemas foram observados no uso da porta USB 2.0 para provisionamento. Para obter resultados consistentes, use a unidade USB incluída com a porta USB 3.0 (ao lado da porta HDMI).

Para o Lenovo ThinkEdge® SE7 0, uma unidade USB não está incluída no aparelho. Use uma unidade USB 3.0 com pelo menos 1 GB de armazenamento.

Configuração do dispositivo

Problema: o dispositivo mostra uma tela em branco durante a inicialização.

Depois de concluir a sequência de inicialização inicial, que leva cerca de um minuto, o dispositivo mostra uma tela em branco por um minuto ou mais enquanto carrega o modelo e inicia a aplicação. Além disso, o dispositivo não gera vídeo se você conectar um monitor depois que ele estiver ligado.

Problema: o dispositivo não responde quando pressiono o botão ligar/desligar para desligá-lo.

O dispositivo leva até 10 segundos para ser desligado com segurança. Você precisa manter o botão ligar/desligar pressionado por apenas 1 segundo para iniciar a sequência de desligamento. Para obter uma lista completa de operações do botão, consulte Botões e luzes do AWS Panorama Appliance.

Problema: preciso gerar um novo arquivo de configuração para alterar as configurações ou substituir um certificado perdido.

AWS Panorama não armazena o certificado do dispositivo ou a configuração de rede depois de baixá-lo e você não pode reutilizar os arquivos de configuração. Exclua o equipamento usando o AWS Panorama console e crie um novo com um novo arquivo de configuração.

Configuração da aplicação

Problema: quando executo várias aplicações, não consigo controlar qual delas usa a saída HDMI.

Quando você implanta várias aplicações que têm nós de saída, a aplicação iniciada mais recentemente usa a saída HDMI. Se esse aplicação parar de ser executada, outra aplicação poderá usar a saída. Para permitir que somente uma aplicação acesse a saída, remova o nó de saída e a borda correspondente do manifesto da outra aplicação e reimplante-os.

Problema: a saída da aplicação não aparece nos logs

Configure um logger em Python para gravar arquivos de log em `/opt/aws/panorama/logs`. Eles são capturados em um fluxo de logs para o nó do contêiner de código. A saída padrão e os streams de erro são capturados em um fluxo de logs separado chamado `console-output`. Se você usar `print`, use a opção `flush=True` para evitar que as mensagens fiquem presas no buffer de saída.

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

Fonte: AWS Panorama serviço

Cada vez que você implanta uma alteração em uma aplicação, registra uma versão de patch que representa a configuração do pacote e os arquivos de ativos de cada pacote que ela usa. Use o script de patches de limpeza para cancelar o registro de versões de patch não utilizadas.

Streams de câmeras

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

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

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

Fonte: log do nó da câmera

O dispositivo não consegue se conectar ao stream da câmera da aplicação. Quando isso acontece, a saída de vídeo fica em branco ou congela no último quadro processado enquanto o aplicativo espera por um quadro de vídeo do SDK do AWS Panorama aplicativo. O software do dispositivo tenta se conectar ao stream da câmera e registra os erros de tempo limite no log do nó da câmera. Verifique se o URL do stream da câmera está correto e se o tráfego RTSP é roteável entre a câmera e o dispositivo em sua rede. Para obter mais informações, consulte Conectar o AWS Panorama Appliance à sua rede.

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

Fonte: log OCC

O AWS Secrets Manager segredo das credenciais do stream da câmera não foi encontrado. Exclua o stream da câmera e recrie-o.

Erro: Camera did not provide an H264 encoded stream

Fonte: log do nó da câmera

O stream da câmera tem uma codificação diferente de H.264, como H.265. Reimplante a aplicação com um stream de câmera H.264. Para obter detalhes sobre as câmeras compatíveis, consulte Câmeras compatíveis.

Segurança no AWS Panorama

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isto como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam ao AWS Panorama, consulte Serviços da AWS em escopo por programa de conformidade.
- Segurança na nuvem: sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o AWS Panorama. Os tópicos a seguir mostram como configurar o AWS Panorama para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros serviços da AWS que ajudam a monitorar e proteger os recursos do AWS Panorama.

Tópicos

Atributos de segurança do AWS Panorama Appliance

Para proteger suas aplicações, modelos e hardware contra códigos maliciosos e outras explorações, o AWS Panorama Appliance implementa um amplo conjunto de atributos de segurança. Eles incluem, mas não estão limitados a.

- Criptografia de disco completo — O equipamento implementa a criptografia de disco inteiro da configuração de chave unificada do Linux (LUKS2). Todos os dados do software do sistema e da aplicação são criptografados com uma chave específica para o seu dispositivo. Mesmo com acesso físico ao dispositivo, um invasor não pode inspecionar o conteúdo do armazenamento.
- Randomização do layout da memória: para se proteger contra ataques direcionados ao código executável carregado na memória, o AWS Panorama Appliance usa a randomização do layout do espaço de endereço (ASLR). A ASLR randomiza a localização do código do sistema operacional à medida que ele é carregado na memória. Isso evita o uso de explorações que tentam sobrescrever ou executar seções específicas do código, prevendo onde elas são armazenadas no runtime.
- Ambiente de execução confiável — O equipamento usa um ambiente de execução confiável (TEE) baseado em ARM TrustZone, com recursos isolados de armazenamento, memória e processamento. As chaves e outros dados confidenciais armazenados na zona de confiança só podem ser acessados por uma aplicação confiável, que é executada em um sistema operacional separado dentro do TEE. O software do AWS Panorama Appliance é executado no ambiente Linux não confiável junto com o código da aplicação. Ele só pode acessar operações criptográficas fazendo uma solicitação à aplicação segura.
- Provisionamento seguro: quando você provisiona um dispositivo, as credenciais (chaves, certificados e outros materiais criptográficos) que você transfere para o dispositivo são válidas apenas por um curto período. O equipamento usa as credenciais de curta duração para se conectar AWS IoT e solicitar um certificado válido por mais tempo. O serviço AWS Panorama gera credenciais e as criptografa com uma chave codificada no dispositivo. Somente o dispositivo que solicitou o certificado pode descriptografá-lo e se comunicar com o AWS Panorama.
- Inicialização segura: quando o dispositivo é inicializado, cada componente do software é autenticado antes de ser executado. A ROM de inicialização, software codificado no processador que não pode ser modificado, usa uma chave de criptografia codificada para descriptografar o carregador de inicialização, que valida o kernel do ambiente de execução confiável e assim por diante.

- Kernel assinado: os módulos do kernel são assinados com uma chave de criptografia assimétrica. O kernel do sistema operacional decifra a assinatura com a chave pública e verifica se ela corresponde à assinatura do módulo antes de carregar o módulo na memória.
- dm-verity: da mesma forma que os módulos do kernel são validados, o dispositivo usa o atributo `dm-verity` do Linux Device Mapper para verificar a integridade da imagem do software do dispositivo antes de montá-la. Se o software do dispositivo for modificado, ele não será executado.
- Prevenção de reversão: quando você atualiza o software do dispositivo, o dispositivo queima um fusível eletrônico no SoC (system on a chip, sistema em um chip). Cada versão do software espera que um número crescente de fusíveis queime e não pode funcionar se mais fusíveis estiverem queimados.

Práticas recomendadas de segurança do AWS Panorama Appliance

Lembre-se das seguintes melhores práticas ao usar o AWS Panorama Appliance.

- Proteja fisicamente o dispositivo: instale o dispositivo em um rack de servidor fechado ou em uma sala segura. Limite o acesso físico ao dispositivo ao pessoal autorizado.
- Proteja a conexão de rede do dispositivo: conecte o dispositivo a um roteador que limite o acesso aos recursos internos e externos. O dispositivo precisa se conectar às câmeras, que podem estar em uma rede interna segura. Ele também precisa se conectar à AWS. Use a segunda porta Ethernet somente para redundância física e configure o roteador para permitir somente o tráfego necessário.

  Use uma das configurações de rede recomendadas para planejar seu layout de rede. Para obter mais informações, consulte Conectar o AWS Panorama Appliance à sua rede.
- Formate a unidade USB: depois de provisionar um dispositivo, remova a unidade USB e formate-a. O dispositivo não usa a unidade USB depois de se registrar no serviço AWS Panorama. Formate a unidade para remover credenciais temporárias, arquivos de configuração e logs de provisionamento.
- Mantenha o dispositivo atualizado: aplique as atualizações do software do dispositivo em tempo hábil. Quando você visualiza um dispositivo no console do AWS Panorama, o console notifica se uma atualização de software está disponível. Para obter mais informações, consulte Gerenciamento de um AWS Panorama Appliance.

  Com a operação da DescribeDeviceAPI, você pode automatizar a verificação de atualizações comparando os `CurrentSoftware` campos `LatestSoftware` e. Quando a versão mais recente do software for diferente da versão atual, aplique a atualização com o console ou usando a CreateJobForDevicesoperação.
- Se você parar de usar um dispositivo, reinicie-o: antes de retirar o dispositivo do seu datacenter seguro, reinicie-o totalmente. Com o dispositivo desligado e conectado, pressione o botão ligar/desligar e o botão de reinicialização simultaneamente por 5 segundos. Isso exclui as credenciais da conta, as aplicações e os logs do dispositivo.

  Para obter mais informações, consulte Botões e luzes do AWS Panorama Appliance.
- Limite o acesso ao AWS Panorama e a outros serviços da AWS — AWSPanoramaFullAccessEle fornece acesso a todas as operações da API do AWS Panorama e, conforme necessário, acesso

a outros serviços. Sempre que possível, a política limita o acesso aos recursos com base nas convenções de nomenclatura. Por exemplo, ele fornece acesso a AWS Secrets Manager segredos que têm nomes começando com `panorama`. Para usuários que precisam de acesso somente para leitura ou acesso a um conjunto mais específico de recursos, use a política gerenciada como ponto de partida para suas políticas de privilégio mínimo.

Para obter mais informações, consulte Políticas do IAM baseadas em identidade para o AWS Panorama.

Proteção de dados no AWS Panorama

O modelo de responsabilidade AWS compartilhada O modelo se aplica à proteção de dados no Panorama da AWS. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Data Privacy FAQ. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o AWS Panorama ou outro Serviços da AWS usando o console AWS CLI, a API ou AWS SDKs. Quaisquer dados inseridos em tags ou em campos de texto de

formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Seções

Criptografia em trânsito

Os endpoints da API do AWS Panorama oferecem suporte a conexões seguras somente em HTTPS. Ao gerenciar recursos do AWS Panorama com o AWS Management Console, o SDK da AWS ou a API do AWS Panorama, toda a comunicação é criptografada com Transport Layer Security (TLS). A comunicação entre o AWS Panorama Appliance e a AWS também é criptografada com TLS. A comunicação entre o AWS Panorama Appliance e as câmeras via RTSP não é criptografada.

Para obter uma lista completa de endpoints de API, consulte Regiões e endpoints da AWS na Referência geral da AWS.

AWS Panorama Appliance

O AWS Panorama Appliance tem portas físicas para Ethernet, vídeo HDMI e armazenamento USB. O slot para cartão SD, o Wi-Fi e o Bluetooth não podem ser usados. A porta USB é usada somente durante o provisionamento para transferir um arquivo de configuração para o dispositivo.

O conteúdo do arquivo de configuração, que inclui o certificado de provisionamento e a configuração de rede do dispositivo, não é criptografado. O AWS Panorama não armazena esses arquivos; eles só podem ser recuperados quando você registra um dispositivo. Depois de transferir o arquivo de configuração para um dispositivo, exclua-o do computador e do dispositivo de armazenamento USB.

Todo o sistema de arquivos do dispositivo é criptografado. Além disso, o dispositivo aplica várias proteções em nível de sistema, incluindo proteção contra reversão para atualizações de software necessárias, kernel assinado e bootloader e verificação da integridade do software.

Ao parar de usar o dispositivo, execute uma redefinição completa para excluir os dados da aplicação e redefinir o software do dispositivo.

Aplicações

Você controla o código que implanta no seu dispositivo. Valide todo o código da aplicação em busca de problemas de segurança antes de implantá-lo, independentemente da origem do código. Se você usa bibliotecas de terceiros em sua aplicação, avalie cuidadosamente as políticas de licenciamento e suporte dessas bibliotecas.

O uso da CPU, da memória e do disco da aplicação não é limitado pelo software do dispositivo. Uma aplicação que usa recursos excessivamente pode afetar negativamente outras aplicações e a operação do dispositivo. Teste as aplicações separadamente antes de combiná-las ou implantá-las em ambientes de produção.

Os ativos da aplicação (códigos e modelos) não estão isolados do acesso em sua conta, dispositivo ou ambiente de compilação. As imagens do contêiner e os arquivos de modelos gerados pela CLI da aplicação do AWS Panorama não são criptografados. Use contas separadas para workloads de produção e permita o acesso apenas conforme a necessidade.

Outros serviços da

Para armazenar seus modelos e contêineres de aplicações com segurança no Amazon S3, o AWS Panorama usa criptografia no lado do servidor com uma chave gerenciada pelo Amazon S3. Para ter mais informações, consulte Como proteger dados usando criptografia no Guia do usuário do Amazon Simple Storage Service.

As credenciais de transmissão da câmera são criptografadas em repouso no AWS Secrets Manager. O perfil do IAM do dispositivo concede a ele permissão para recuperar o segredo a fim de acessar o nome de usuário e a senha do stream.

O AWS Panorama Appliance envia dados de log para o Amazon CloudWatch Logs. CloudWatch O Logs criptografa esses dados por padrão e pode ser configurado para usar uma chave gerenciada pelo cliente. Para obter mais informações, consulte Criptografar dados de log em CloudWatch registros usando AWS KMS o Guia do usuário do Amazon CloudWatch Logs.

Gerenciamento de identidade e acesso para o AWS Panorama

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (fazer login) e autorizado (ter permissões) para usar recursos do AWS Panorama. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz no Panorama da AWS.

Usuário do serviço: se você usa o serviço AWS Panorama para fazer o trabalho, seu administrador fornece as credenciais e as permissões necessárias. À medida que usar mais atributos do AWS Panorama para fazer seu trabalho, você poderá precisar de permissões adicionais. Compreenda como o acesso é gerenciado pode ajudar a solicitar as permissões corretas ao administrador. Se você não puder acessar um atributo no AWS Panorama, consulte Solução de problemas de identidade e acesso do AWS Panorama.

Administrador do serviço: se você for o responsável pelos recursos do AWS Panorama na empresa, provavelmente terá acesso total ao AWS Panorama. Cabe a você determinar quais atributos e recursos do AWS Panorama os usuários do serviço deverão acessar. Envie as solicitações ao

administrador do IAM para alterar as permissões dos usuários de serviço. Revise as informações nesta página para compreender os conceitos básicos do IAM. Para saber mais sobre como a empresa pode usar o IAM com o AWS Panorama, consulte Como o AWS Panorama funciona com o IAM.

Administrador do IAM: se você for um administrador do IAM, talvez queira saber detalhes sobre como é possível criar políticas para gerenciar o acesso ao AWS Panorama. Para visualizar exemplos de políticas baseadas em identidade do AWS Panorama que podem ser usadas no IAM, consulte Exemplos de políticas baseadas em identidade do AWS Panorama.

Autenticar com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para designar solicitações por conta própria, consulte Versão 4 do AWS Signature para solicitações de API no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser necessário fornecer informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no Guia do usuário do AWS IAM Identity Center e Usar a autenticação multifator da AWS no IAM no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte Tarefas que exigem credenciais de usuário-raiz no Guia do Usuário do IAM.

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, é recomendável contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, é recomendável alternar as chaves de acesso. Para obter mais informações, consulte Alternar as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Casos de uso para usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Para assumir temporariamente uma função do IAM no AWS Management Console, você pode alternar de um usuário para uma função do IAM (console). Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para usar perfis, consulte Métodos para assumir um perfil no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, é possível criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidade de terceiros (federação) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no Guia do Usuário do AWS IAM Identity Center .
- Permissões temporárias para usuários do IAM: um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal da chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.
  - Perfil de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de

serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

- Aplicativos em execução na Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo solicitações AWS CLI de AWS API. Isso é preferível ao armazenamento de chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte Usar uma função do IAM para conceder permissões a aplicativos executados em EC2 instâncias da Amazon no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a

ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o AWS WAF Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte Políticas de controle de serviços no Guia AWS Organizations do Usuário.
- Políticas de controle de recursos (RCPs) — RCPs são políticas JSON que você pode usar para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as políticas do IAM anexadas a cada recurso que você possui. O RCP limita as permissões para recursos nas contas dos membros e pode afetar as permissões efetivas para identidades, incluindo a Usuário raiz da conta da AWS, independentemente de pertencerem à sua organização. Para obter mais informações sobre Organizations e RCPs, incluindo uma lista Serviços da AWS desse suporte RCPs, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do

usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como o AWS Panorama funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao AWS Panorama, você precisa saber quais atributos do IAM estão disponíveis para uso com o AWS Panorama. Para ter uma visão de alto nível de como o AWS Panorama e outros AWS serviços funcionam com o IAM, consulte AWS os serviços que funcionam com o IAM no Guia do usuário do IAM.

Para obter uma visão geral de permissões, políticas e funções como são usadas pelo AWS Panorama, consulte AWS Panorama permissões.

Exemplos de políticas baseadas em identidade do AWS Panorama

Por padrão, os usuários e as funções do IAM não têm permissão para criar ou modificar recursos do AWS Panorama. Eles também não podem realizar tarefas usando a AWS API AWS Management Console AWS CLI, ou. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e perfis permissão para executarem operações de API específicas nos recursos especificados de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM utilizando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Tópicos

- Práticas recomendadas de política
- Uso do console do AWS Panorama

- Permitir que os usuários visualizem suas próprias permissões

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do AWS Panorama em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter

mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Uso do console do AWS Panorama

Para acessar o console do AWS Panorama, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do Panorama AWS em sua AWS conta. Se você criar uma política baseada em identidade que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis do IAM) com essa política.

Para ter mais informações, consulte Políticas do IAM baseadas em identidade para o AWS Panorama

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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 gerenciadas para o AWS Panorama

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque elas estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para obter mais informações, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

O AWS Panorama fornece as seguintes políticas gerenciadas. Para ver o conteúdo completo e o histórico de alterações de cada política, consulte as páginas vinculadas no console do IAM.

- AWSPanoramaFullAccess— Fornece acesso total ao AWS Panorama, aos pontos de acesso do AWS Panorama no Amazon S3, às credenciais do dispositivo e aos registros do dispositivo na

AWS Secrets Manager Amazon. CloudWatch Inclui permissão para criar um perfil vinculado ao serviço para o AWS Panorama.

- AWSPanoramaServiceLinkedRolePolicy— Permite que o AWS Panorama gerencie recursos no AWS IoT, no AWS Secrets Manager e no AWS Panorama.
- AWSPanoramaApplianceServiceRolePolicy— Permite que um AWS Panorama Appliance carregue registros e obtenha objetos dos pontos de acesso do Amazon S3 criados pelo AWS Panorama. CloudWatch

Atualizações do AWS Panorama em políticas AWS gerenciadas

A tabela a seguir descreve as atualizações das políticas gerenciadas para o AWS Panorama.

Alteração	Descrição	Data
AWSPanoramaApplianceServiceRolePolicy — Atualização de uma política existente	StringLike Substitua a condição ArnLike por para escrever ARNs.	2024-12-10
AWSPanoramaFullAccess — Atualização de uma política existente	StringLike Substitua a condição ArnLike por para escrever ARNs.	2024-12-10
AWSPanoramaFullAccess — Atualização de uma política existente	Foram adicionadas permissões à política do usuário para permitir que os usuários visualizem grupos de CloudWatch registros no console de registros.	2022-01-13
AWSPanoramaFullAccess — Atualização de uma política existente	Permissões adicionadas à política de usuário para permitir que os usuários gerenciem a função vinculada ao serviço AWS Panorama e acessem recursos do AWS Panorama em outros serviços,	2021-10-20

Alteração	Descrição	Data
	incluindo IAM, Amazon S3 e Secrets CloudWatch Manager.	
AWSPanoramaApplianceServiceRolePolicy — Nova política	Nova política para o perfil de serviço do AWS Panorama Appliance	2021-10-20
AWSPanoramaServiceLinkedRolePolicy — Nova política	Nova política para o perfil vinculado a serviço AWS Panorama.	2021-10-20
O AWS Panorama começou a monitorar alterações	O AWS Panorama começou a monitorar as mudanças em suas políticas AWS gerenciadas.	2021-10-20

Usando funções vinculadas a serviços para AWS Panorama

AWS Panorama usa funções vinculadas ao serviço AWS Identity and Access Management (IAM). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente a. AWS Panorama As funções vinculadas ao serviço são predefinidas AWS Panorama e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Uma função vinculada ao serviço facilita a configuração AWS Panorama porque você não precisa adicionar manualmente as permissões necessárias. AWS Panorama define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido de outra forma, só AWS Panorama pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, e essa política não pode ser anexada a nenhuma outra entidade do IAM.

Uma função vinculada ao serviço poderá ser excluída somente após excluir seus recursos relacionados. Isso protege seus recursos do AWS Panorama , pois você não pode remover por engano as permissões de acesso aos recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas aos serviços, consulte Serviços da AWS que funcionam com o IAM e procure os serviços que apresentam Sim na coluna Funções vinculadas aos serviços. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Seções

Permissões de função vinculadas ao serviço para AWS Panorama

AWS Panorama usa a função vinculada ao serviço chamada AWSServiceRoleForAWSPanorama—Permite que o AWS Panorama gerencie recursos no AWS IoT, no AWS Secrets Manager e no AWS Panorama.

A função AWSService RoleFor AWSPanorama vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `panorama.amazonaws.com`

A política de permissões de função AWS Panorama permite concluir as seguintes ações:

- Monitore AWS Panorama os recursos
- Gerencie AWS IoT recursos para o AWS Panorama equipamento
- Acesse AWS Secrets Manager segredos para obter as credenciais da câmera

Para ver uma lista completa de permissões, veja a AWSPanorama ServiceLinkedRolePolicy política no console do IAM.

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para obter mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Criação de uma função vinculada ao serviço para AWS Panorama

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você registra um dispositivo na AWS Management Console, na ou na AWS API AWS CLI, AWS Panorama cria a função vinculada ao serviço para você.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando você registra um equipamento, AWS Panorama cria a função vinculada ao serviço para você novamente.

Editando uma função vinculada ao serviço para AWS Panorama

AWS Panorama não permite que você edite a função AWSService RoleFor AWSPanorama vinculada ao serviço. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluindo uma função vinculada ao serviço para AWS Panorama

Se você não precisar mais usar um atributo ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de seu perfil vinculado ao serviço antes de excluí-lo manualmente.

Para excluir os AWS Panorama recursos usados pelo AWSService RoleForAWSPanorama, use os procedimentos nas seções a seguir deste guia.

- Excluir versões e aplicações
- Cancelamento do registro de um dispositivo

> **Note**
>
> Se o AWS Panorama serviço estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir a função AWSService RoleFor AWSPanorama vinculada ao serviço, use o console do IAM AWS CLI, o ou a AWS API. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões suportadas para funções vinculadas a AWS Panorama serviços

AWS Panorama suporta o uso de funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte Regiões e endpoints da AWS.

Prevenção contra o ataque do “substituto confuso” em todos os serviços

“Confused deputy” é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com diretores de serviços que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição `aws:SourceAccount`global `aws:SourceArn`e as chaves de contexto nas políticas de recursos para limitar as permissões que AWS Panorama concedem outro serviço ao recurso. Se você utilizar ambas as chaves de contexto de condição global, o valor `aws:SourceAccount` e a conta `aws:SourceArn` no valor deverão utilizar o mesmo ID de conta quando utilizados na mesma instrução de política.

O valor de `aws:SourceArn` deve ser o ARN de um AWS Panorama dispositivo.

A maneira mais eficaz de se proteger do problema ‘confused deputy’ é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou se especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com curingas (`*`) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

Para obter instruções sobre como proteger a função de serviço AWS Panorama usada para dar permissão ao AWS Panorama equipamento, consulte. Proteção do perfil do dispositivo

Solução de problemas de identidade e acesso do AWS Panorama

Use as seguintes informações para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com o AWS Panorama e o IAM.

Tópicos

- Não tenho autorização para executar uma ação no AWS Panorama
- Não estou autorizado a realizar iam: PassRole
- Desejo permitir que pessoas fora da minha conta da AWS acessem meus recursos do AWS Panorama

Não tenho autorização para executar uma ação no AWS Panorama

Se isso AWS Management Console indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. O administrador é a pessoa que forneceu o seu nome de usuário e senha.

O erro de exemplo a seguir ocorre quando o usuário `mateojackson` do IAM tenta usar o console para visualizar detalhes sobre um dispositivo, mas não tem permissões `panorama:DescribeAppliance`.

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

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso `my-appliance` usando a ação `panorama:DescribeAppliance`.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não tem autorização para executar a ação `iam:PassRole`, as suas políticas deverão ser atualizadas para permitir a passagem de um perfil para o AWS Panorama.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro exemplificado a seguir ocorre quando um usuário do IAM chamado `marymajor` tenta usar o console para executar uma ação no AWS Panorama. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Desejo permitir que pessoas fora da minha conta da AWS acessem meus recursos do AWS Panorama

É possível criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o AWS Panorama oferece suporte a esses atributos, consulte Como o AWS Panorama funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Validação de conformidade do AWS Panorama

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Governança e conformidade de segurança: esses guias de implementação de solução abordam considerações sobre a arquitetura e fornecem etapas para implantar recursos de segurança e conformidade.
- Referência de serviços qualificados para HIPAA: lista os serviços qualificados para HIPAA. Nem todos Serviços da AWS são elegíveis para a HIPAA.
- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Guias de conformidade do cliente — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a Referência de controles do Security Hub.
- Amazon GuardDuty — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca

de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.

- AWS Audit Manager— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Considerações adicionais sobre quando pessoas estão presentes

Abaixo estão algumas das melhores práticas a serem consideradas ao usar o AWS Panorama para cenários em que pessoas possam estar presentes:

- Certifique-se de que você esteja ciente e em conformidade com todas as leis e regulamentações aplicáveis ao seu caso de uso. Isso pode incluir leis relacionadas ao posicionamento e ao campo de visão de suas câmeras, requisitos de aviso e sinalização ao colocar e usar câmeras e os direitos das pessoas que possam estar presentes em seus vídeos, incluindo os direitos delas à privacidade.
- Leve em consideração o efeito de suas câmeras nas pessoas e na privacidade delas. Além dos requisitos legais, considere se seria apropriado colocar um aviso nas áreas onde suas câmeras estão localizadas e se as câmeras devem ser colocadas à vista de todos e sem quaisquer obstáculos, para que as pessoas não se surpreendam com o fato de serem filmadas.
- Tenha políticas e procedimentos apropriados para a operação de suas câmeras e para a análise dos dados obtidos das câmeras.
- Considere controles de acesso, limitações de uso e períodos de retenção apropriados para os dados obtidos de suas câmeras.

Segurança da infraestrutura no AWS Panorama

Como um serviço gerenciado, o AWS Panorama é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar o AWS Panorama por meio da rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Implantação do AWS Panorama Appliance em seu datacenter

O AWS Panorama Appliance precisa de acesso à Internet para se comunicar com AWS os serviços. Ele também precisa acessar sua rede interna de câmeras. É importante considerar cuidadosamente a configuração da sua rede e fornecer a cada dispositivo apenas o acesso necessário. Tenha cuidado se sua configuração permitir que o AWS Panorama Appliance atue como uma ponte para uma rede confidencial de câmeras IP.

Você é responsável por fazer o seguinte:

- A segurança física e lógica da rede do AWS Panorama Appliance.
- Operar com segurança as câmeras conectadas à rede ao usar o AWS Panorama Appliance.
- Mantendo o AWS Panorama Appliance e o software da câmera atualizados.
- Cumprir todas as leis ou regulamentações aplicáveis associadas ao conteúdo dos vídeos e imagens que você coleta em seus ambientes de produção, incluindo aqueles relacionados à privacidade.

O AWS Panorama Appliance usa streams de câmera RTSP não criptografados. Para obter mais informações sobre como conectar o AWS Panorama Appliance à sua rede, consulte Conectar o AWS Panorama Appliance à sua rede. Para obter detalhes sobre criptografia, consulte Proteção de dados no AWS Panorama.

Software de ambiente de runtime no AWS Panorama

O AWS Panorama fornece um software que executa o código do sua aplicação em um ambiente baseado em Ubuntu Linux no AWS Panorama Appliance. O AWS Panorama é responsável por manter atualizado o software na imagem do dispositivo. O AWS Panorama lança regularmente atualizações de software, que você pode aplicar usando o console do AWS Panorama.

Você pode usar bibliotecas no código da sua aplicação instalando-as no `Dockerfile` da aplicação. Para garantir a estabilidade da aplicação em todas as compilações, escolha uma versão específica de cada biblioteca. Atualize suas dependências regularmente para resolver problemas de segurança.

Versões

A tabela a seguir mostra quando os recursos e as atualizações de software foram lançados para o AWS Panorama serviço, o software e a documentação. Para garantir que você tenha acesso a todos os recursos, atualize seu AWS Panorama aparelho para a versão mais recente do software. Para obter mais informações, consulte o tópico vinculado.

Alteração	Descrição	Data
Aviso de fim do suporte	Aviso de fim do suporte: em 31 de maio de 2026, AWS encerrará o suporte para AWS Panorama. Depois de 31 de maio de 2026, você não poderá mais acessar o AWS Panorama console ou os AWS Panorama recursos. Para obter mais informações, consulte AWS Panorama Fim do suporte.	20 de maio de 2025
Atualização das políticas gerenciadas	AWS Identity and Access Management as políticas gerenciadas para AWS Panorama foram atualizadas. Para obter detalhes, consulte Políticas gerenciadas pela AWS.	10 de dezembro de 2024
Atualização do software do dispositivo	A versão 7.0.13 é uma atualização de versão principal que altera a forma como o equipamento gerencia as atualizações de software. Se você restringir a comunicação de rede de saída do dispositivo ou conectá-lo a uma sub-	28 de dezembro de 2023

	rede VPC privada, deverá permitir o acesso a endpoints e portas adicionais antes de aplicar a atualização. Para obter mais informações, consulte o log de alterações.	
Atualização do software do dispositivo	A versão 6.2.1 inclui correções de bugs. Para obter mais informações, consulte o log de alterações.	6 de setembro de 2023
Atualização do software do dispositivo	A versão 6.0.8 inclui correções de bugs e melhorias de segurança. Para obter mais informações, consulte o log de alterações.	6 de julho de 2023
Atualização do software do dispositivo	A versão 5.1.7 inclui correções de bugs e melhorias no tratamento de erros. Para obter mais informações, consulte o log de alterações.	31 de março de 2023
Atualização do console	Agora você pode comprar o AWS Panorama equipamento no console de gerenciamento. Para conceder permissão ao usuário para comprar dispositivos, consulte as políticas do IAM baseadas em identidade para o AWS Panorama.	2 de fevereiro de 2023
Atualização do software do dispositivo	A versão 5.0.74 inclui correções de bugs e melhorias no tratamento de erros. Para obter mais informações, consulte o log de alterações.	23 de janeiro de 2023

Atualização da API	Adição da opção `AllowMajorVersionUpdate` a `OTAJobConfig` para tornar as atualizações da versão principal do software do dispositivo sejam opcionais. Para obter mais informações, consulte CreateJobForDevices.	19 de janeiro de 2023
Nova ferramenta para desenvolvedores	Uma nova ferramenta, “sideloading”, está disponível no repositório de amostras. AWS Panorama GitHub Você pode usar essa ferramenta para atualizar o código da aplicação sem criar e implantar um contêiner. Para obter mais informações, consulte o README.	16 de novembro de 2022
Atualização da imagem base da aplicação	A versão 1.2.0 adiciona uma opção de tempo limite a `video_in.get()` , define a variável de ambiente `AWS_REGION` e melhora o tratamento de erros. Para obter mais informações, consulte o log de alterações.	16 de novembro de 2022
Atualização do software do dispositivo	A versão 5.0.42 inclui correções de bugs e atualizações de segurança. Para obter mais informações, consulte o log de alterações.	16 de novembro de 2022

Atualização do software do dispositivo	A versão 5.0.7 adiciona suporte para reinicializar dispositivos remotamente e pausar streams de câmera remotamente. Para obter mais informações, consulte o log de alterações.	13 de outubro de 2022
Atualização do software do dispositivo	A versão 4.3.93 adiciona suporte para recuperar logs de um dispositivo offline. Para obter mais informações, consulte o log de alterações.	24 de agosto de 2022
Atualização do software do dispositivo	A versão 4.3.72 inclui correções de bugs e atualizações de segurança. Para obter mais informações, consulte o log de alterações.	23 de junho de 2022
AWS PrivateLink apoio	AWS Panorama oferece suporte a endpoints VPC para gerenciar AWS Panorama recursos de uma sub-rede privada. Para obter mais informações, consulte Usar endpoints da VPC.	2 de junho de 2022
Atualização do software do dispositivo	A versão 4.3.55 melhora a utilização do armazenamento do log `console_output` . Para obter mais informações, consulte o log de alterações.	05 de maio de 2022

Lenovo ThinkEdge® 0 SE7	Um novo aparelho para AWS Panorama está disponível na Lenovo. O Lenovo ThinkEdge ® SE7 0, equipado com Nvidia Jetson Xavier NX, suporta os mesmos recursos do Appliance. AWS Panorama Para obter mais informaçõ es, consulte Dispositivos compatíveis.	6 de abril de 2022
Atualização da imagem base da aplicação	A versão 1.1.0 melhora o desempenho ao executar threads em segundo plano e adiciona um sinalizador (is_cached) aos objetos de mídia para indicar se a imagem está atualizada. Para obter mais informações, consulte gallery.ecr.aws.	29 de março de 2022
Atualização do software do dispositivo	A versão 4.3.45 adiciona suporte para acesso à GPU e portas de entrada. Para obter mais informações, consulte o log de alterações.	24 de março de 2022
Atualização do software do dispositivo	A versão 4.3.35 melhora a segurança e o desempenho. Para obter mais informações, consulte o log de alterações.	22 de fevereiro de 2022

Atualização das políticas gerenciadas	AWS Identity and Access Management as políticas gerenciadas para AWS Panorama foram atualizadas. Para obter detalhes, consulte Políticas gerenciadas pela AWS.	13 de janeiro de 2022
Logs de provisionamento	Com o software 4.3.23, o dispositivo grava logs em uma unidade USB durante o provisionamento. Para obter mais informações, consulte Logs.	13 de janeiro de 2022
Configuração do servidor NTP	Agora você pode configurar o AWS Panorama equipamento para usar um servidor NTP específico para sincronização do relógio. Defina as configurações de NTP durante a configuração do dispositivo com outras configurações de rede. Para obter mais informações, consulte Configurar.	13 de janeiro de 2022
Regiões adicionais	AWS Panorama agora está disponível nas regiões Ásia-Pacífico (Cingapura) e Ásia-Pacífico (Sydney).	13 de janeiro de 2022

Atualização do software do dispositivo	A versão 4.3.4 adiciona suporte para a configuração `precisionMode` de modelos e atualiza o comportamento de registro em log. Para obter mais informações, consulte o log de alterações.	8 de novembro de 2021
Atualização das políticas gerenciadas	AWS Identity and Access Management as políticas gerenciadas para AWS Panorama foram atualizadas. Para obter detalhes, consulte Políticas gerenciadas pela AWS.	20 de outubro de 2021
Disponibilidade geral	AWS Panorama agora está disponível para todos os clientes nas regiões Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon), Europa (Irlanda) e Canadá (Central). Para comprar um AWS Panorama eletrodoméstico, visite AWS Panorama.	20 de outubro de 2021
Demonstração	AWS Panorama está disponível por convite nas regiões Leste dos EUA (Norte da Virgínia) e Oeste dos EUA (Oregon).	1º de dezembro de 2020