Manual do usuário

AWS IoT Analytics

AWS IoT Analytics: Manual do usuário

Table of Contents

O que é o AWS IoT Analytics? 1
Como usar o AWS IoT Analytics 1
Recursos principais 2
Componentes e conceitos do AWS IoT Analytics 4
Acessar AWS IoT Analytics 7
Casos de uso 8
Conceitos básicos (console) 9
Faça login no AWS IoT Analytics console 10
Criar um canal 10
Criar um datastore 12
Criar um pipeline 13
Criar um conjunto de dados 15
Envie dados da mensagem com AWS IoT 17
Verifique o progresso das AWS IoT mensagens 18
Acessar resultados da consulta 19
Explorar seus dados 19
Modelos de cadernos 22
Conceitos básicos 23
Criar um canal 23
Criação de um datastore 25
Políticas do Amazon S3 25
Formatos de arquivo 27
Partições personalizadas 30
Criando um pipeline 33
Consumir dados para o AWS IoT Analytics 34
Usando o agente de mensagens AWS IoT 35
Uso da API BatchPutMessage 39
Monitorando os dados ingeridos 40
Criação de um conjunto de dados 42
Consultar dados 43
Acessando os dados consultados 43
Explorar dados AWS IoT Analytics 19
Amazon S3 44
AWS IoT Events 45

Amazon QuickSight 45
Bloco de anotações Jupyter 46
Mantendo várias versões dos conjuntos de dados 46
Sintaxe da carga útil da mensagem 47
Trabalho com dados AWS IoT SiteWise 48
Criar um conjunto de dados 48
Acessar o conteúdo do conjunto de dados 52
Tutorial: consultar AWS IoT SiteWise dados 54
Atividades do pipeline 62
Atividade Canal 62
Atividade Datastore 62
Atividade AWS Lambda 63
Exemplo 1 da função do Lambda 63
Exemplo 2 da função do Lambda 66
Atividade AddAttributes 67
Atividade RemoveAttributes 68
Atividade SelectAttributes 69
Atividade Filtro 70
Atividade DeviceRegistryEnrich 70
Atividade DeviceShadowEnrich 72
Atividade Matemática 74
Funções e operadores de atividades matemáticas 75
RunPipelineActivity 92
Reprocessamento de mensagens do canal 94
Parâmetros 94
Reprocessar mensagens do canal (console) 95
Reprocessamento de mensagens do canal (API) 96
Cancelamento de atividades de reprocessamento de canais 97
Automação de seu fluxo de trabalho 98
Casos de uso 99
Como usar um contêiner do Docker 100
Variáveis de entrada/saída do contêiner docker personalizado 103
Permissões 105
CreateDataset (Java e AWS CLI) 107
Exemplo 1: criação de um conjunto de dados SQL (java) 108
Exemplo 2: criação de um conjunto de dados SQL com uma janela delta (java) 109

Exemplo 3: criação de um conjunto de dados de contêiner com seu próprio trigger de programação (java) ... 110
Exemplo 4: criação de um conjunto de dados de contêiner com um conjunto de dados SQL como um trigger (java) ... 111
Exemplo 5: criação de um conjunto de dados SQL (CLI) ... 112
Exemplo 6: criação de um conjunto de dados SQL com uma janela delta (CLI) ... 112
Conteinerização de caderno ... 114
Habilitar a conteinerização de instâncias de cadernos não criadas pelo console AWS IoT Analytics ... 115
Atualizar a extensão de conteinerização do notebook ... 117
Criar uma imagem conteinerizada ... 117
Usando um contêiner personalizado ... 123
Visualizando dados ... 132
Visualizando (console) ... 132
Visualização (QuickSight) ... 133
Marcação ... 137
Conceitos básicos de tags ... 137
Utilização de tags com políticas do IAM ... 138
Restrições de tags ... 140
Expressões SQL ... 142
Funcionalidade SQL compatível ... 143
Tipos de dados compatíveis ... 143
Funções compatíveis ... 144
Solução de problemas comuns ... 145
Segurança ... 146
AWS Identity and Access Management ... 146
Público ... 146
Autenticando com identidades ... 147
Gerenciamento de acesso ... 150
Trabalhando com IAM ... 152
Prevenção contra o ataque “Confused deputy” em todos os serviços ... 157
Exemplos de política do IAM ... 163
Solução de problemas de identidade e acesso ... 169
Registro e monitoramento ... 171
Ferramentas de monitoramento automatizadas ... 171
Ferramentas de monitoramento manual ... 171

Monitorar com o CloudWatch Logs 172
Monitoramento com CloudWatch Events 177
Registrar em log chamadas de API com o CloudTrail 186
Validação de conformidade 191
Resiliência 192
Segurança da infraestrutura 192
Cotas 194
Comandos 195
Ações AWS IoT Analytics 195
Dados do AWS IoT Analytics 195
Solução de problemas 196
Como saber se minhas mensagens estão chegando no AWS IoT Analytics? 196
Por que meu pipeline perde mensagens? Como posso corrigir isso? 197
Por que não há dados em meu datastore? 198
Por que meu conjunto de dados simplesmente mostra `__dt`? 198
Como fazer para codificar um evento orientado pela conclusão do conjunto de dados? 199
Como fazer para configurar corretamente minha instância de caderno para usar o AWS IoT Analytics? 199
Por que não consigo criar cadernos em uma instância? 199
Por que não estou vendo meus conjuntos de dados no Amazon QuickSight? 200
Por que não vejo o botão conteinerizar em meu caderno Jupyter existente? 200
Por que minha instalação do plug-in de conteinerização está falhando? 201
Por que meu plug-in de conteinerização está emitindo um erro? 201
Por que não vejo minhas variáveis durante a conteinerização? 201
Quais variáveis posso adicionar a meu contêiner como uma entrada? 202
Como faço para definir a saída de meu contêiner como uma entrada para a análise subsequente? 202
Por que meu conjunto de dados de contêiner está falhando? 202
Histórico do documento 203
Atualizações anteriores 204
..... ccv

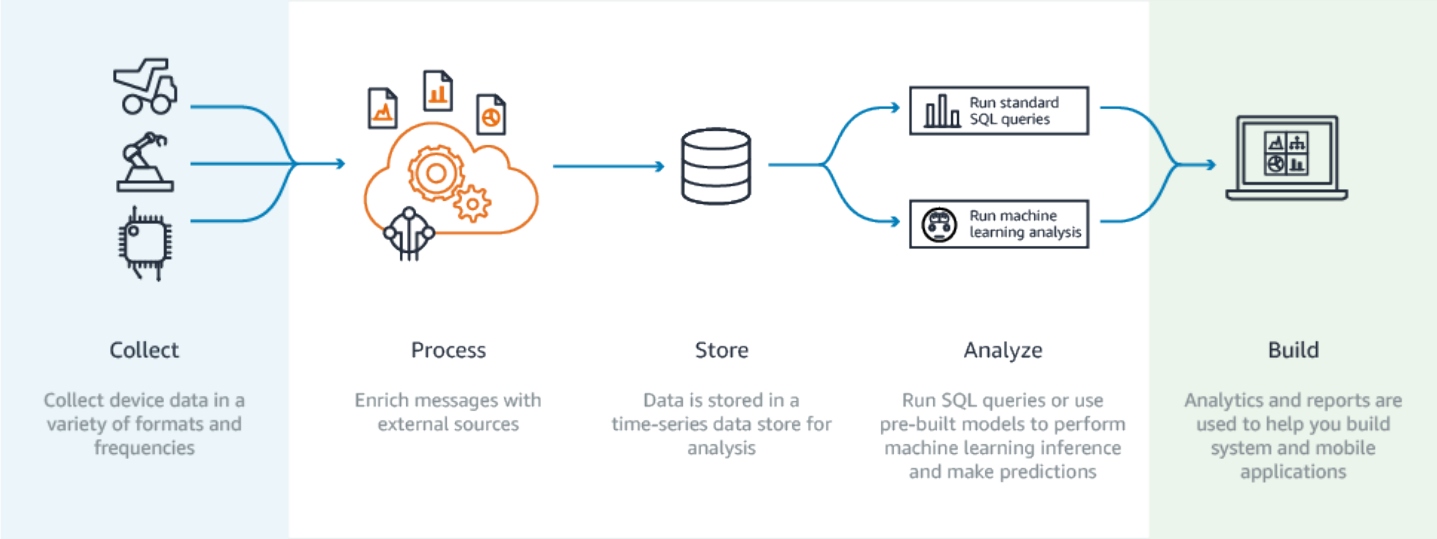
O que é o AWS IoT Analytics?

AWS IoT Analytics automatiza as etapas necessárias para analisar dados de dispositivos de IoT. AWS IoT Analytics filtra, transforma e enriquece os dados de IoT antes de armazená-los em um armazenamento de dados de séries temporais para análise. É possível configurar o serviço para coletar somente os dados que você precisa nos dispositivos, aplicar transformações matemáticas para processar os dados e enriquecê-los com metadados específicos do dispositivo, tais como tipo e localização do dispositivo, antes de armazenar os dados processados. Em seguida, você pode analisar seus dados executando consultas usando o mecanismo de consulta SQL integrado ou realizar análises mais complexas e inferências de machine learning. AWS IoT Analytics permite a exploração avançada de dados por meio da integração com o caderno Jupyter. AWS IoT Analytics também permite a visualização de dados por meio da integração com o Amazon QuickSight. O Amazon QuickSight está disponível nas seguintes regiões.

As tradicionais ferramentas de análise e inteligência de negócios são projetadas para processar dados estruturados. Os dados brutos da IoT normalmente vêm de dispositivos que registram dados menos estruturados (como temperatura, movimento ou som). Como resultado, os dados desses dispositivos podem ter lacunas significativas, mensagens corrompidas e leituras falsas que devem ser limpas antes que a análise ocorra. Além disso, os dados de IoT geralmente só são significativos no contexto de outros dados de fontes externas. AWS IoT Analytics permite que você resolva esses problemas e colete grandes quantidades de dados do dispositivo, processe mensagens e as armazene. Em seguida, você pode consultar os dados e analisá-los. AWS IoT Analytics inclui modelos pré-criados para casos de uso comuns de IoT para que você possa responder a perguntas, como quais dispositivos estão prestes a falhar ou quais clientes demonstram probabilidade de abandonar seus dispositivos wearables.

Como usar o AWS IoT Analytics

O gráfico a seguir mostra uma visão geral de como você pode usar AWS IoT Analytics.



Recursos principais

Coletar

- Integrado com o AWS IoT Core — AWS IoT Analytics está totalmente integrado com AWS IoT Core para que ele receba mensagens de dispositivos conectados à medida que são transmitidas.
- Use uma API de lotes para adicionar dados de qualquer origem – AWS IoT Analytics pode receber dados de qualquer origem por meio de HTTP. Isto significa que qualquer dispositivo ou serviço que está conectado à Internet pode enviar dados para AWS IoT Analytics. Para obter mais informações, consulte BatchPutMessage em Referência de API AWS IoT Analytics.
- Colete somente os dados que você deseja armazenar e analisar — você pode usar o console AWS IoT Analytics para configurar AWS IoT Analytics o recebimento de mensagens de dispositivos por meio de filtros de tópicos do MQTT em vários formatos e frequências. AWS IoT Analytics valida se os dados estão dentro dos parâmetros específicos que você define e cria canais. Em seguida, o serviço encaminha os canais para pipelines apropriados, para processamento, transformação e enriquecimento de mensagens.

Processar

- Limpar e filtrar — AWS IoT Analytics permite definir funções do AWS Lambda que são acionadas quando o AWS IoT Analytics detecta dados ausentes, para que você possa executar códigos para estimar e preencher lacunas. Você também pode definir filtros máximos e mínimos e limites percentuais para remover exceções de seus dados.

- Transformar — AWS IoT Analytics pode transformar as mensagens usando lógica matemática ou condicional definidas, para que você possa realizar cálculos comuns como a conversão de Celsius em Fahrenheit.
- Enriquecer — AWS IoT Analytics pode enriquecer os dados com fontes de dados externas, como previsão do tempo e, em seguida, rotear os dados para o datastore do AWS IoT Analytics.

Armazene

- Datastore de séries temporais — AWS IoT Analytics armazena os dados do dispositivo em um datastore de séries temporais otimizado para uma recuperação e análise mais rápidas. Também é possível gerenciar permissões de acesso, implementar políticas de retenção de dados e exportar seus dados para pontos de acesso externos.
- Armazenar dados processados e brutos — AWS IoT Analytics armazena os dados processados e também armazena automaticamente os dados brutos incluídos para que você possa processá-los posteriormente.

Analisar

- Executar consultas SQL ad-hoc — AWS IoT Analytics fornece um mecanismo de consulta SQL para que você possa executar consultas ad-hoc e obter resultados rapidamente. O serviço habilita o uso de consultas SQL padrão para extrair dados do datastore para responder a perguntas como qual a distância média percorrida por uma frota de veículos conectados ou quantas portas são trancadas após as 19h em um edifício inteligente. Essas consultas podem ser reutilizadas mesmo se os dispositivos conectados, o tamanho da frota e os requisitos analíticos forem alterados.
- Análise de séries temporais — AWS IoT Analytics é compatível com análises de séries temporais para que você possa analisar o desempenho de dispositivos ao longo do tempo e entender como e onde eles estão sendo usados, monitorar continuamente os dados do dispositivo para fazer previsões de problemas de manutenção e monitorar sensores para prever e reagir às condições ambientais.
- Notebooks hospedados para análise sofisticada e machine learning — AWS IoT Analytics inclui suporte para notebooks hospedados no caderno Jupyter para análise estatística e machine learning. O serviço inclui um conjunto de modelos de caderno que contêm modelos e AWS visualizações de machine learning criados por eles. Você pode usar os modelos para iniciar os casos de uso de IoT relacionados ao perfil de falha do dispositivo, fazendo previsão de eventos como baixa utilização, que pode sinalizar que o cliente deixará de usar o produto, ou segmentando dispositivos por níveis de uso do cliente (por exemplo, usuários regulares, usuários de finais de semana) ou integridade do dispositivo. Depois de criar um caderno, você

pode conteinerizá-lo e executá-lo em uma programação especificada por você. Para obter mais informações, consulte Automação de seu fluxo de trabalho.

- Previsão — Você pode fazer uma classificação estatística por meio de um método chamado de regressão logística. Você também pode usar a Long-Short-Term Memory (LSTM – Memória de longo a curto prazo), que é uma poderosa técnica de rede neural para fazer previsões de saída ou do estado de um processo que varia com o tempo. Os modelos de blocos de anotações pré-criados também são compatíveis com o algoritmo de clustering K-means para segmentação de dispositivo, que agrupa seus dispositivos em grupos de dispositivos semelhantes. Esses modelos normalmente são usados para o perfil de integridade e de estado de dispositivos, como unidades de HVAC em uma fábrica de chocolate ou desgaste de lâminas em uma turbina eólica. Novamente, esses modelos de caderno podem ser conteinerizados e executados em uma programação.

Criar e visualizar

- Integração do Amazon QuickSight — o AWS IoT Analytics fornece um conector para o Amazon QuickSight para que você possa visualizar seus conjuntos de dados em um painel do QuickSight.
- Integração do console — também é possível visualizar os resultados ou a análise ad-hoc no caderno Jupyter incorporado no console do AWS IoT Analytics.

Componentes e conceitos do AWS IoT Analytics

Canal

Um canal coleta dados de um tópico MQTT e arquiva as mensagens brutas não processadas antes de publicar os dados em uma pipeline. Você também pode enviar mensagens diretamente para um canal usando a API BatchPutMessage. As mensagens não processadas são armazenadas em um bucket do Amazon Simple Storage Service (Amazon S3) gerenciado por você ou AWS IoT Analytics.

Pipeline

Um pipeline consome mensagens de um canal e permite que você as processe antes de armazená-las em um datastore. As etapas de processamento, chamadas de atividades (Atividades de pipeline), executam transformações em suas mensagens, como a remoção, a renomeação ou a adição de atributos a mensagens, filtrando-as com base em valores de atributos, invocando funções do Lambda em mensagens para processamento avançado ou executando transformações matemáticas para normalizar dados de dispositivos.

Datastore

Os pipelines armazenam as mensagens processadas em um datastore. Um datastore não é apenas um banco de dados; é um repositório escalável e consultável de suas mensagens. Você pode ter vários armazenamentos de dados para mensagens provenientes de diferentes dispositivos ou locais, ou filtradas por atributos de mensagens de acordo com a configuração e os requisitos do pipeline. Assim como ocorre com mensagens de canal não processadas, as mensagens processadas do datastore são armazenadas em um bucket do Amazon S3 gerenciado por você ou AWS IoT Analytics.

Conjunto de dados

Você recupera dados de um armazenamento de dados criando um conjunto de dados. AWS IoT Analytics permite criar um conjunto de dados SQL ou um conjunto de dados de contêiner.

Quando tiver um conjunto de dados, você poderá explorar e obter informações sobre seus dados usando a integração ao Amazon QuickSight. Ou você pode executar funções de análise mais avançadas por meio da integração ao caderno Jupyter. O caderno Jupyter fornece poderosas ferramentas de ciência de dados que podem realizar machine learning e uma ampla variedade de análises estatísticas. Para obter mais informações, consulte Modelos de caderno.

É possível enviar conteúdo do conjunto de dados para um bucket do Amazon S3, permitindo a integração com os data lakes existentes ou o acesso usando aplicativos internos e ferramentas de visualização. Também é possível enviar o conteúdo do conjunto de dados como uma entrada para o AWS IoT Events, um serviço que permite monitorar dispositivos ou processos para procurar falhas ou alterações na operação e para acionar ações adicionais quando esses eventos ocorrerem.

Conjunto de dados SQL

Um conjunto de dados SQL é semelhante a uma visualização materializada de um banco de dados SQL. Você pode criar um conjunto de dados SQL com a aplicação de uma ação SQL. Os conjuntos de dados SQL podem ser gerados automaticamente em uma programação recorrente por meio da especificação de um trigger.

Conjunto de dados de contêiner

Um conjunto de dados de contêiner habilita que você execute automaticamente suas ferramentas de análise e gere resultados. Para obter mais informações, consulte Automação de seu fluxo de trabalho. Reúne um conjunto de dados SQL como entrada, um contêiner de Docker com suas ferramentas de análise e arquivos de bibliotecas necessárias, variáveis de entrada e saída e um

trigger de programação opcional. As variáveis de entrada e saída informam à imagem executável onde obter os dados e armazenar os resultados. O trigger pode executar sua análise quando um conjunto de dados SQL conclui a criação de seu conteúdo ou de acordo com uma expressão de cronograma. Um conjunto de dados de contêiner executa, gera e salva automaticamente os resultados das ferramentas de análise.

Trigger

Você pode criar automaticamente um conjunto de dados especificando um trigger. O gatilho pode ser um intervalo de tempo (por exemplo, criar esse conjunto de dados a cada duas horas) ou quando o conteúdo de outro conjunto de dados foi criado (por exemplo, criar esse conjunto de dados quando a criação do conteúdo de `myOtherDataset` for concluída). Ou você pode gerar conteúdo do conjunto de dados manualmente usando a API CreateDatasetContent.

Contêiner de docker

É possível criar seu próprio contêiner do Docker para empacotar suas ferramentas de análise ou usar opções que o SageMaker fornece. Para obter mais informações, consulte Contêiner do Docker. É possível criar seu próprio contêiner do Docker para empacotar suas ferramentas de análise ou usar opções fornecidas pelo SageMaker. É possível armazenar um contêiner em um registro do Amazon ECR especificado por você para que ele esteja disponível para instalação na plataforma desejada. Os contêineres do Docker podem executar seu código de análise personalizada preparado com Matlab, Octave, Wise.io, SPSS, R, Fortran, Python, Scala, Java, C ++ e assim por diante. Para obter mais informações, consulte Conteinerização de um caderno.

Janelas delta

Janelas delta são uma série de períodos definidos pelo usuário, intervalos não sobrepostos e contíguos. As janelas delta habilitam a criação de conteúdo de conjunto de dados e a execução de análise de dados novos recebidos no datastore desde a última análise. Você cria uma janela delta configurando o `deltaTime` na parte `filters` de uma `queryAction` de um conjunto de dados. Para obter mais informações, consulte a API `CreateDataset`. Geralmente, o conteúdo do conjunto de dados é criado automaticamente ao configurar também um gatilho de intervalo de tempo (`triggers:schedule:expression`). Isso permite que você filtre as mensagens que chegaram durante um período específico, para que os dados contidos nas mensagens dos períodos anteriores não sejam contados duas vezes. Para obter mais informações, consulte Exemplo 6: criando um conjunto de dados SQL com uma janela delta (CLI).

Acessar AWS IoT Analytics

Como parte do AWS IoT, AWS IoT Analytics fornece as seguintes interfaces para permitir que seus dispositivos gerem dados e os aplicativos interajam com os dados gerados por eles:

AWS Command Line Interface (AWS CLI)

Execute comandos para a AWS IoT Analytics no Windows, OS X e Linux. Esses comandos permitem que você crie e gerencie coisas, certificados, regras e políticas. Para começar a usar, consulte o Guia do usuário do AWS Command Line Interface. Para obter mais informações sobre comandos da AWS IoT, consulte iot no Referência AWS Command Line Interface.

> **Important**
>
> Use o comando `aws iotanalytics` para interagir com AWS IoT Analytics. Use o comando `aws iot` para interagir com outras partes do sistema IoT.

AWS IoT API

Crie seus aplicativos para IoT usando solicitações HTTP ou HTTPS. Essas ações de API permitem que você crie e gerencie coisas, certificados, regras e políticas. Para obter mais informações, consulte Ações do na Referência de API do AWS IoT.

AWS SDKs

Crie seus aplicativos AWS IoT Analytics usando APIs específicas de uma linguagem. Esses SDKs encapsulam a API HTTP e HTTPS e permitem que você programe em qualquer uma das linguagens suportadas. Para obter mais informações, consulte AWS SDKs e ferramentas.

SDKs de dispositivo da AWS IoT

Crie aplicativos para serem executados em seus dispositivos que enviam para o AWS IoT Analytics. Para obter mais informações, consulte AWS IoT SDKs.

Console do AWS IoT Analytics

Você pode criar os componentes para visualizar os resultados no console AWS IoT Analytics.

Casos de uso

Manutenção preditiva

O AWS IoT Analytics fornece modelos para criar modelos de manutenção preditiva e aplicá-los aos dispositivos. Por exemplo, é possível usar o AWS IoT Analytics para prever quando os sistemas de aquecimento e resfriamento provavelmente apresentarão falha nos veículos de carga conectados, para que os veículos possam ser reencaminhados para evitar danos à remessa. Ou um fabricante de automóveis pode detectar quais de seus clientes estão com as pastilhas de freio gastas e alertá-los para fazer manutenção em seus veículos.

Reabastecimento proativo de suprimentos

O AWS IoT Analytics permite criar aplicativos para IoT que podem monitorar inventários em tempo real. Por exemplo, uma empresa do setor de alimentos e bebidas pode analisar os dados de máquinas de vendas de alimentos e reordenar de maneira proativa as mercadorias sempre que os suprimentos estiverem acabando.

Pontuação de eficiência do processo

Com o AWS IoT Analytics, você pode criar aplicativos de IoT que monitoram constantemente a eficiência de diferentes processos e realizam ações para melhorar o processo. Por exemplo, uma empresa do setor de mineração pode aumentar a eficiência de seus caminhões de minério maximizando a carga para cada viagem. Com o AWS IoT Analytics, a empresa pode identificar a carga mais eficiente para um local ou caminhão ao longo do tempo e, em seguida, comparar quaisquer desvios da carga pretendida em tempo real e planejar melhor as diretrizes de carregamento para melhorar a eficiência.

Agricultura inteligente

O AWS IoT Analytics pode enriquecer os dados do dispositivo IoT com metadados contextuais usando dados do registro do AWS IoT ou fontes de dados públicas para que sua análise calcule o tempo, o local, a temperatura, a altitude e outras condições ambientais. Com essa análise, você pode escrever modelos que resultam em ações recomendadas para seus dispositivos seguirem. Por exemplo, para determinar quando molhar as plantas, os sistemas de irrigação podem enriquecer os dados do sensor de umidade com dados de precipitação, permitindo um uso mais eficiente da água.

Introdução ao AWS IoT Analytics (console)

Use este tutorial para criar os AWS IoT Analytics recursos (também conhecidos como componentes) necessários para descobrir informações úteis sobre os dados do seu dispositivo de IoT.

> Observações
>
> - Se você inserir caracteres maiúsculos no tutorial a seguir, altere-os AWS IoT Analytics automaticamente para minúsculas.
> - O AWS IoT Analytics console tem um recurso de introdução com um clique para criar um canal, pipeline, armazenamento de dados e conjunto de dados. Você pode encontrar esse atributo ao entrar no console do AWS IoT Analytics .
>   - Este tutorial orienta você em cada etapa da criação de seus AWS IoT Analytics recursos.

Siga as instruções abaixo para criar um AWS IoT Analytics canal, pipeline, armazenamento de dados e conjunto de dados. O tutorial também mostra como usar o AWS IoT Core console para enviar mensagens que serão ingeridas. AWS IoT Analytics

Tópicos

- Faça login no AWS IoT Analytics console
- Criar um canal
- Criar um datastore
- Criar um pipeline
- Criar um conjunto de dados
- Envie dados da mensagem com AWS IoT
- Verifique o progresso das AWS IoT mensagens
- Acessar resultados da consulta
- Explorar seus dados
- Modelos de cadernos

Faça login no AWS IoT Analytics console

Para começar, você precisa ter uma AWS conta. Se você já tiver uma AWS conta, navegue até https://console.aws.amazon.com/iotanalytics/.

Se você não tiver uma AWS conta, siga estas etapas para criar uma.

Para criar uma AWS conta

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções on-line.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

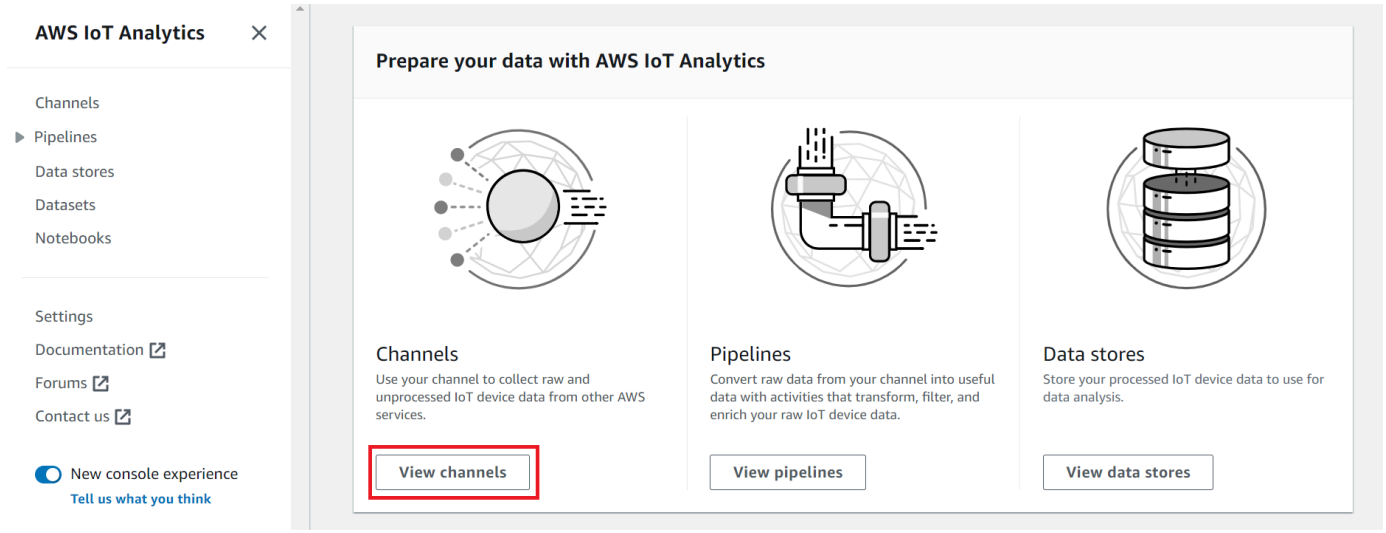
   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como uma prática recomendada de segurança, atribua o acesso administrativo para um usuário e use somente o usuário-raiz para executar tarefas que requerem o acesso de usuário-raiz.
3. Faça login no AWS Management Console e navegue até https://console.aws.amazon.com/iotanalytics/.

Criar um canal

Um canal coleta e arquiva dados brutos, não processados e não estruturados de dispositivos de IoT. Siga estas etapas para criar seu canal.

Para criar um canal

1. Em https://console.aws.amazon.com/iotanalytics/, na seção Preparar seus dados com AWS IoT Analytics, escolha Visualizar canais.



> **Tip**
>
> Você também pode escolher Canais no painel de navegação.

2. Na página Channels (Canais), escolha Create channel (Criar canal).
3. Na página Especificar detalhes do canal, insira os detalhes do seu canal.
   a. Insira um nome de canal que seja exclusivo e que você possa identificar facilmente.
   b. (Opcional) Em Tags, adicione uma ou mais tags personalizadas (pares chave-valor) ao seu canal. As tags ajudam a identificar os recursos que você cria para AWS IoT Analytics.
   c. Escolha Próximo.
4. AWS IoT Analytics armazena seus dados brutos e não processados do dispositivo de IoT em um bucket do Amazon Simple Storage Service (Amazon S3). Você pode escolher seu próprio bucket do Amazon S3, que você pode acessar e gerenciar, ou AWS IoT Analytics pode gerenciar o bucket do Amazon S3 para você.
   a. Neste tutorial, em Tipo de armazenamento, escolha Armazenamento gerenciado pelo serviço.
   b. Em Escolher por quanto tempo armazenar seus dados brutos, escolha Indefinidamente.
   c. Escolha Próximo.
5. Na página Configurar fonte, insira as informações das quais AWS IoT Analytics coletar dados da mensagem AWS IoT Core.

a. Insira um filtro de AWS IoT Core tópico, por exemplo,`update/environment/dht1`. Posteriormente neste tutorial, você usará esse filtro de tópicos para enviar dados de mensagens para o seu canal.

b. Na área Nome do perfil do IAM, escolha Criar novo. Na janela Criar nova função, insira um nome para a função e selecione Criar função. Isso cria automaticamente uma função com uma política adequada anexada a ela.

c. Escolha Próximo.

6. Verifique suas escolhas e selecione Criar canal.
7. Verifique se seu novo canal aparece na página Canais.

Criar um datastore

Um datastore recebe e armazena os dados de suas mensagens. Um datastore não é um banco de dados. Um datastore é um repositório escalável e consultável em um bucket do Amazon S3. É possível usar vários datastores para mensagens que chegam de diferentes dispositivos ou locais. Outra opção é filtrar os dados das mensagens de acordo com a configuração e os requisitos do pipeline.

Siga estas etapas para criar um datastore.

Para criar um datastore

1. Em https://console.aws.amazon.com/iotanalytics/, na seção Preparar seus dados com AWS IoT Analytics, escolha Visualizar datastores.
2. Na página Datastores, selecione Criar datastore.
3. Na página Especificar detalhes do datastore, insira informações básicas sobre seu datastore.

   a. Em ID do datastore, insira uma ID exclusiva do datastore. Você não pode alterar a ID depois de criá-la.

   b. (Opcional) Em Tags, escolha Adicionar nova tag para adicionar uma ou mais tags personalizadas (pares chave-valor) ao seu datastore. As tags ajudam a identificar os recursos que você cria para AWS IoT Analytics.

   c. Escolha Próximo.

4. Na página Configurar tipo de armazenamento, especifique como armazenar seus dados.

a. Em Tipo de armazenamento, escolha Armazenamento gerenciado pelo serviço.

b. Em Configurar quanto tempo você deseja manter seus dados processados, escolha Indefinidamente.

c. Escolha Next (Próximo).

5. AWS IoT Analytics os armazenamentos de dados oferecem suporte aos formatos de arquivo JSON e Parquet. Para o formato de dados do seu datastore, escolha JSON ou Parquet. Consulte Formatos de arquivo para obter mais informações sobre os tipos de AWS IoT Analytics com suporte.

   Escolha Próximo.

6. (Opcional) AWS IoT Analytics oferece suporte a partições personalizadas em seu armazenamento de dados para que você possa consultar dados eliminados para melhorar a latência. Para obter mais informações sobre partições personalizadas compatíveis, consulte Partições personalizadas.

   Escolha Próximo.

7. Verifique suas escolhas e selecione Criar datastore.

8. Verifique se seu novo datastore aparece na página Datastores.

Criar um pipeline

Você deve criar um pipeline para conectar um canal a um datastore. Um pipeline básico especifica apenas o canal que coleta os dados e identifica o datastore para o qual as mensagens são enviadas. Para obter mais informações, consulte Atividades do pipeline.

Neste tutorial, você cria um pipeline que conecta somente um canal a um datastore. Posteriormente, você pode adicionar atividades de pipeline para processar esses dados.

Siga estas etapas para criar um pipeline.

Para criar um pipeline

1. Em https://console.aws.amazon.com/iotanalytics/, na seção Preparar seus dados com AWS IoT Analytics, escolha Visualizar pipelines.

> **Tip**
>
> Você também pode escolher Pipelines no painel de navegação.

2. Na página Pipelines, selecione Criar pipeline.
3. Insira os detalhes do seu pipeline.
   a. Em Configurar ID e fontes do pipeline, insira o nome do pipeline.
   b. Escolha a fonte do seu funil, que é um AWS IoT Analytics canal do qual seu funil lerá as mensagens.
   c. Especifique a saída do seu pipeline, o datastore em que os dados da mensagem processada são armazenados.
   d. (Opcional) Em Tags, adicione uma ou mais tags personalizadas (pares chave-valor) ao seu pipeline.
   e. Na página Inferir atributos da mensagem, insira um nome de atributo e um valor de exemplo, escolha um tipo de dados na lista e escolha Adicionar atributo.
   f. Repita a etapa anterior para todos os atributos necessários e, em seguida, escolha Próximo.
   g. Não será adicionada nenhuma atividade de pipeline no momento. Portanto, na página Enriquecer, transformar e filtrar mensagens, basta selecionar Próximo.
4. Verifique suas escolhas e selecione Criar pipeline.
5. Verifique se seu novo pipeline aparece na página Pipelines.

> **Note**
>
> Você criou AWS IoT Analytics recursos para que eles possam fazer o seguinte:
>
> - Coletar dados brutos e não processados de mensagens de dispositivos de IoT com um canal.
> - Armazenar os dados de mensagens do seu dispositivo de IoT em um datastore.
> - Limpe, filtre, transforme e enriqueça seus dados com um pipeline.
>
> Em seguida, você criará um conjunto de dados AWS IoT Analytics SQL para descobrir informações úteis sobre seu dispositivo de IoT.

Criar um conjunto de dados

> **Note**
>
> Um conjunto de dados geralmente é uma coleção de dados que podem ou não estar organizados em formato tabular. Por outro lado, AWS IoT Analytics cria seu conjunto de dados aplicando uma consulta SQL aos dados em seu armazenamento de dados.

Agora você tem um canal que roteia os dados brutos da mensagem para um pipeline que os armazena em um datastore no qual eles podem ser consultados. Para consultar os dados, crie um conjunto de dados. Um conjunto de dados contém instruções e expressões SQL usadas para consultar o datastore juntamente com uma programação adicional que repete a consulta em um dia e horário que você especifica. Você pode usar expressões semelhantes às expressões de CloudWatch agendamento da Amazon para criar os horários opcionais.

Para criar um conjunto de dados

1. Em https://console.aws.amazon.com/iotanalytics/, no painel de navegação esquerdo, escolha Conjuntos de dados.
2. Na página Criar conjunto de dados, escolha Criar SQL.
3. Na página Especificar detalhes do conjunto de dados, especifique os detalhes do seu conjunto de dados.

   a. Digite um nome para o conjunto de dados.

   b. Em Fonte do datastore, escolha a ID exclusiva que identifica o datastore que você criou anteriormente.

   c. (Opcional) Em Tags, adicione uma ou mais tags personalizadas (pares chave-valor) ao seu conjunto de dados.
4. Use expressões SQL para consultar seus dados e responder perguntas analíticas. Os resultados da sua consulta são armazenados nesse conjunto de dados.

   a. No campo Consulta do autor, insira uma consulta SQL que usa um curinga para mostrar até cinco linhas de dados.

```
SELECT * FROM my_data_store LIMIT 5
```

Para obter mais informações sobre a funcionalidade SQL suportada em AWS IoT Analytics, consulteExpressões SQL em AWS IoT Analytics.

b. Você pode escolher Consulta de teste para validar se sua entrada está correta e exibir os resultados em uma tabela após a consulta.

> **Note**
>
> - Neste ponto do tutorial, seu datastore deve estar vazio. A execução de uma consulta SQL em um datastore vazio não retornará resultados, então talvez você veja apenas `__dt`.
> - Tenha o cuidado de limitar sua consulta SQL a um tamanho razoável para que ela não seja executada por um longo período, pois o Athena limita o número máximo de consultas em execução. Por isso, você deve ter o cuidado de limitar a consulta SQL a um tamanho razoável.
>
> Sugerimos usar uma cláusula de `LIMIT` em sua consulta durante o teste. Depois que o teste for bem-sucedido, você poderá remover essa cláusula.

5. (Opcional) Quando você cria o conteúdo do conjunto de dados usando dados de um período especificado, alguns dados podem não chegar a tempo de serem processados. Para permitir um atraso, você pode especificar um deslocamento ou delta. Para ter mais informações, consulte Receber notificações de dados atrasadas por meio do Amazon CloudWatch Events.

   Você não configurará um filtro de seleção de dados neste momento. Na página Configurar filtro de seleção de dados, escolha Próximo.

6. (Opcional) Você pode programar essa consulta para ser executada regularmente para atualizar o conjunto de dados. As programações de conjuntos de dados podem ser criadas e editadas a qualquer momento.

   Uma execução recorrente da consulta não será programada neste momento. Portanto, na página Definir programação de consulta, selecione Próximo.

7. AWS IoT Analytics criará versões desse conteúdo do conjunto de dados e armazenará seus resultados de análise pelo período especificado. Recomendamos 90 dias, mas você pode optar por definir sua política de retenção personalizada. Você também pode limitar o número de versões armazenadas do conteúdo do seu conjunto de dados.

Você pode usar o período de retenção padrão do conjunto de dados como Indefinidamente e manter o Versionamento desativado. Na página Configurar os resultados da sua análise, escolha Próximo.

8. (Opcional) Você pode configurar as regras de entrega dos resultados do seu conjunto de dados para um destino específico, como AWS IoT Events.

   Você não fornecerá seus resultados em nenhum outro lugar neste tutorial. Portanto, na página Configurar regras de entrega de conteúdo do conjunto de dados, escolha Próximo.
9. Verifique suas escolhas e selecione Criar conjunto de dados.
10. Verifique se seu novo conjunto de dados aparece na página Conjuntos de dados.

Envie dados da mensagem com AWS IoT

Se você tem um canal que roteia dados para um pipeline que armazena os dados em um datastore onde eles podem ser consultados, está pronto para enviar dados de mensagem para o AWS IoT Analytics. Você pode enviar dados AWS IoT Analytics usando as seguintes opções:

- Use o mediador de AWS IoT mensagens.
- Use a operação de API AWS IoT Analytics BatchPutMessage.

Nas etapas a seguir, você envia dados de mensagens do agente de AWS IoT mensagens no AWS IoT Core console para que ele AWS IoT Analytics possa ingerir esses dados.

> **Note**
>
> Ao criar nomes de tópicos para suas mensagens, observe o seguinte:
>
> - Os nomes de tópicos diferenciam maiúsculas de minúsculas. Campos denominados `example` e `EXAMPLE` na mesma carga útil são considerados duplicatas.
> - Os nomes dos tópicos não podem começar com o caractere $. Os nomes de tópicos que começam com $ são tópicos reservados e serão usados somente pelo AWS IoT.
> - Não inclua informações de identificação pessoal nos nomes dos tópicos, pois essas informações podem aparecer em comunicações e relatórios não criptografados.
> - AWS IoT Core não consigo enviar mensagens entre AWS contas ou AWS regiões.

Para enviar dados de mensagens com AWS IoT

1. Faça login no console do AWS IoT.
2. No painel de navegação, escolha Teste e, em seguida, escolha Cliente de teste MQTT.
3. No Cliente de teste MQTT, escolha Publicar em um tópico.
4. Em Nome do tópico, insira um nome que corresponda ao filtro de tópico que você inseriu ao criar um canal. Este exemplo usa `update/environment/dht1`.
5. Em Carga útil da mensagem, insira o conteúdo do JSON a seguir:

```
{
  "thingid": "dht1",
  "temperature": 26,
  "humidity": 29,
  "datetime": "2018-01-26T07:06:01"
}
```

6. (Opcional) Escolha Adicionar configuração para obter mais opções de protocolo de mensagens.
7. Selecione Publish.

   Isso publica uma mensagem capturada pelo seu canal. Em seguida, seu pipeline encaminha a mensagem para seu datastore.

Verifique o progresso das AWS IoT mensagens

É possível verificar se as mensagens estão sendo ingeridas no canal seguindo estas etapas:

Para verificar o progresso das AWS IoT mensagens

1. Faça login no https://console.aws.amazon.com/iotanalytics/.
2. No painel de navegação, escolha Canais e, em seguida, selecione o nome do canal criado anteriormente.
3. Na página Detalhes do canal, role para baixo até a seção Monitoramento e ajuste o prazo exibido (1h 3h 12h 1d 3d 1s). Escolha um valor como 1s para ver os dados da última semana.

Você pode usar um atributo semelhante para monitorar os erros e o runtime da atividade do pipeline na página Detalhes do pipeline. Neste tutorial, você não especificou atividades como parte do pipeline, então não deve ver nenhum erro de runtime.

Para monitorar a atividade do pipeline

1. No painel de navegação, selecione Pipelines e selecione o nome do pipeline criado anteriormente.
2. Na página Detalhes do pipeline, role para baixo até a seção Monitoramento e ajuste o prazo exibido escolhendo um dos indicadores do prazo (1h 3h 12h 1d 3d 1s).

Acessar resultados da consulta

O conteúdo do conjunto de dados é um arquivo que contém o resultado da consulta no formato CSV.

1. Em https://console.aws.amazon.com/iotanalytics/, no painel de navegação esquerdo, escolha Conjuntos de dados.
2. Na página Conjuntos de dados, escolha o nome do conjunto de dados criado anteriormente:
3. Na página de informações do conjunto de dados, no canto superior direito, selecione Executar agora.
4. Para verificar se o conjunto de dados está pronto, procure no conjunto de dados uma mensagem semelhante a Você iniciou com sucesso a consulta do seu conjunto de dados. A guia Conteúdo do conjunto de dados contém os resultados da consulta e exibe Bem-sucedido.
5. Para visualizar os resultados da sua consulta bem-sucedida, na guia Conteúdo do conjunto de dados, selecione o nome da consulta. Selecione Fazer download para visualizar ou salvar o arquivo CSV que contém os resultados da consulta.

> **Note**
>
> AWS IoT Analytics pode incorporar a parte HTML de um Jupyter Notebook na página de conteúdo do conjunto de dados. Para ter mais informações, consulte Visualizando dados AWS IoT Analytics com o console.

Explorar seus dados

Você tem diversas opções para armazenar, analisar e visualizar seus dados.

Amazon Simple Storage Service

É possível enviar conteúdo do conjunto de dados para um bucket do Amazon S3, permitindo a integração com os data lakes existentes ou o acesso usando aplicativos internos e ferramentas de visualização. Veja o campo `contentDeliveryRules::destination::s3DestinationConfiguration` na CreateDatasetoperação.

AWS IoT Events

Você pode enviar o conteúdo do conjunto de dados como entrada para AWS IoT Events um serviço que permite monitorar dispositivos ou processos em busca de falhas ou alterações na operação e iniciar ações adicionais quando esses eventos ocorrerem.

Para fazer isso, crie um conjunto de dados usando a CreateDatasetoperação e especifique uma AWS IoT Events entrada no campo`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. Você também deve especificar `roleArn` a função, que concede AWS IoT Analytics permissões para execução`iotevents:BatchPutMessage`. Sempre que o conteúdo do conjunto de dados for criado, AWS IoT Analytics enviará cada entrada de conteúdo do conjunto de dados como uma mensagem para a entrada especificada AWS IoT Events . Por exemplo, se seu conjunto de dados contém o seguinte conteúdo.

```
"what","who","dt"
"overflow","sensor01","2019-09-16 09:04:00.000"
"overflow","sensor02","2019-09-16 09:07:00.000"
"underflow","sensor01","2019-09-16 11:09:00.000"
...
```

Em seguida, AWS IoT Analytics envia mensagens que contêm campos como os seguintes.

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

Você desejará criar uma AWS IoT Events entrada que reconheça os campos nos quais você está interessado (um ou mais dos `what`,`who`,`dt`) e criar um modelo de AWS IoT Events detector que use esses campos de entrada em eventos para acionar ações ou definir variáveis internas.

Bloco de anotações Jupyter

O caderno Jupyter é uma solução de código aberto que usa linguagens de desenvolvimento de scripts para análises avançadas e exploração de dados ad-hoc. Você pode se aprofundar e aplicar análises mais complexas e usar métodos de machine learning, como agrupamento k-means e modelos de regressão e clustering para previsão, nos dados do seu dispositivo de IoT.

AWS IoT Analytics usa instâncias de SageMaker notebooks da Amazon para hospedar seus notebooks Jupyter. Antes de criar uma instância de notebook, você deve criar um relacionamento entre AWS IoT Analytics e a Amazon SageMaker:

1. Navegue até o SageMaker console e crie uma instância de notebook:
   a. Preencha os detalhes e, em seguida, selecione Create a new role (Criar uma nova função). Anote o ARN da função.
   b. Crie uma instância de bloco de anotações.
2. Acesse o console do IAM e modifique a SageMaker função:
   a. Abra a função. Ela deve ter uma política gerenciada.
   b. Selecione Adicionar política em linha e, em seguida, em Serviço, selecione iotAnalytics. Selecione Selecionar ações, insira **`GetDatasetContent`** na caixa de pesquisa e selecione. Escolha Revisar política.
   c. Revise a política para verificar sua precisão, insira um nome e, em seguida, selecione Criar política.

Isso dá à função recém-criada permissão para ler um conjunto de dados de AWS IoT Analytics.

1. Volte para https://console.aws.amazon.com/iotanalytics/ e, no painel de navegação esquerdo, escolha Cadernos. Na página Cadernos, selecione Criar:
2. Na página Selecionar um modelo, escolha Modelo em branco IoTA.
3. Na página Configurar caderno, insira um nome para o caderno. Em Selecionar origens de conjunto de dados, escolha Selecionar e, em seguida, selecione o conjunto de dados criado anteriormente. Em Selecionar uma instância do notebook, escolha a instância do notebook em que você criou SageMaker.
4. Depois de revisar suas opções, escolha Criar caderno.
5. Na página Notebooks, sua instância de notebook será aberta no SageMaker console da Amazon.

Modelos de cadernos

Os modelos de AWS IoT Analytics caderno contêm modelos e visualizações de aprendizado de máquina AWS criados por você para ajudar você a começar a usar casos de AWS IoT Analytics uso. Você pode usar esses modelos de caderno para saber mais ou reutilizá-los de acordo com os dados do seu dispositivo de IoT e agregar valor imediato.

Você pode encontrar os seguintes modelos de caderno no AWS IoT Analytics console:

- Detecção de anomalias contextuais: aplicativo da detecção contextual de anomalias na velocidade medida do vento com um modelo de média móvel ponderada exponencialmente de Poisson (PEWMA).
- Previsão de emissão de painéis solares: aplicativo de modelos de série temporal linear, estacional e em partes para previsão da emissão de painéis solares.
- Manutenção preditiva em motores a jato: aplicativo de redes neurais multivariadas de memória de longo prazo (LSTM) e regressão logística para prever falhas em motores a jato.
- Segmentação de clientes de casa inteligente: aplicativo de análise PCA e k-means para detecção de diferentes segmentos de clientes em dados de utilização de casas inteligentes.
- Previsão de congestionamento em cidades inteligentes: aplicativo de LSTM para prever as taxas de utilização de rodovias municipais.
- Previsão de qualidade do ar em cidades inteligentes: aplicativo de LSTM para prever poluição particulada em centros urbanos.

Conceitos básicos do AWS IoT Analytics

Esta seção discute os comandos básicos usados para coletar, armazenar, processar e consultar os dados do dispositivo usando o AWS IoT Analytics. Os exemplos mostrados aqui usam o AWS Command Line Interface (AWS CLI). Para obter mais informações em AWS CLI, consulte o Guia do usuário AWS Command Line Interface. Para obter mais informações sobre os comandos de CLI disponíveis para AWS IoT, consulte a Referência de AWS Command Line Interface.

> **Important**
>
> Use o comando `aws iotanalytics` para interagir com o AWS IoT Analytics usando a AWS CLI. Use o comando `aws iot` para interagir com outras partes do sistema IoT usando a AWS CLI.

> **Note**
>
> Ao digitar os nomes das entidades do AWS IoT Analytics (canal, conjunto de dados, datastore e pipeline) nos exemplos a seguir, lembre-se de que todas as letras maiúsculas usadas serão alteradas automaticamente para minúsculas pelo sistema. Os nomes de entidades devem começar com uma letra minúscula e conter apenas letras minúsculas, sublinhados e dígitos.

Criar um canal

Um canal coleta e arquiva dados de mensagens brutos não processados antes de publicar esses dados em um pipeline. As mensagens recebidas são enviadas a um canal, então, a primeira etapa é criar um canal para os dados.

```
aws iotanalytics create-channel --channel-name mychannel
```

Se quiser que as mensagens do AWS IoT sejam ingeridas no AWS IoT Analytics, é possível criar uma regra Rules Engine do AWS IoT para enviar as mensagens a esse canal. Isso será mostrado posteriormente em Consumir dados para o AWS IoT Analytics. Outra maneira de colocar os dados em um canal é usar o comando `BatchPutMessage` do AWS IoT Analytics.

Para listar os canais que você já criou:

```
aws iotanalytics list-channels
```

Para obter mais informações sobre um canal:

```
aws iotanalytics describe-channel --channel-name mychannel
```

As mensagens do canal não processadas são armazenadas em um bucket do Amazon S3 gerenciado pelo AWS IoT Analytics ou em um gerenciado por você. Use o parâmetro `channelStorage` para especificar qual deles. O padrão é um bucket do Amazon S3 gerenciado pelo serviço. Se você decidir que as mensagens do canal devem ser armazenadas em um bucket do Amazon S3 gerenciado por você, é necessário conceder ao AWS IoT Analytics permissão para executar estas ações no bucket do Amazon S3 em seu nome: `s3:GetBucketLocation` (verificação do local do bucket), `s3:PutObject` (armazenamento), `s3:GetObject` (leitura), `s3:ListBucket` (reprocessamento).

Example

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "MyStatementSid",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::my-iot-analytics-bucket",
                "arn:aws:s3:::my-iot-analytics-bucket/*"
            ]
        }
    ]
```

```
}
```

Se você fizer alterações nas opções ou nas permissões do armazenamento do canal gerenciado pelo cliente, poderá ser necessário reprocessar os dados do canal para garantir que os dados ingeridos anteriormente estejam incluídos no conteúdo do conjunto de dados. Consulte Reprocessar dados do canal.

Criação de um datastore

Um datastore recebe e armazena suas mensagens. Não é um banco de dados; é um repositório escalável e consultável de suas mensagens. Você pode criar vários armazenamentos de dados para armazenar mensagens provenientes de dispositivos ou locais diferentes, ou você pode usar um único armazenamento de dados para receber todas as suas mensagens AWS IoT.

```
aws iotanalytics create-datastore --datastore-name mydatastore
```

Para listar os datastores que você já criou.

```
aws iotanalytics list-datastores
```

Para obter mais informações sobre um datastore.

```
aws iotanalytics describe-datastore --datastore-name mydatastore
```

Políticas do Amazon S3 para recursos AWS IoT Analytics

Você pode armazenar as mensagens do datastore processadas em um bucket do Amazon S3 gerenciado pelo AWS IoT Analytics ou em um gerenciado por você. Ao criar um datastore, selecione o bucket do Amazon S3 que você deseja usando o parâmetro da API `datastoreStorage`. O padrão é um bucket do Amazon S3 gerenciado pelo serviço.

Se você decidir que as mensagens do datastore devem ser armazenadas em um bucket do Amazon S3 gerenciado por você, é necessário conceder ao permissão AWS IoT Analytics para executar estas ações no bucket do Amazon S3 em seu nome:

- `s3:GetBucketLocation`
- `s3:PutObject`
- `s3:DeleteObject`

Se você usar o datastore como uma fonte para um conjunto de dados de consulta SQL, será necessário configurar uma política de bucket do Amazon S3 que concede ao AWS IoT Analytics permissão para invocar consultas do Amazon Athena no conteúdo do bucket.

> **Note**
>
> Recomendamos que você especifique `aws:SourceArn` em sua política de bucket para ajudar a evitar o problema de segurança substituto confuso. Isso restringe o acesso ao permitir somente as solicitações provenientes de uma conta específica. Para obter mais informações sobre o problema substituto confuso, consulte the section called “Prevenção contra o ataque “Confused deputy” em todos os serviços”.

Veja a seguir um exemplo de política de bucket que concede estas permissões necessárias.

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "MyStatementSid",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
```

```
                        "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-
EXAMPLE-DATASET",
                        "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-
EXAMPLE-DATASTORE"
                    ]
                }
            }
        }
    ]
}
```

Consulte Acesso entre contas no Guia do usuário do Amazon Athena para obter mais informações.

> **Note**
>
> Se você atualizar as opções ou permissões do datastore gerenciado pelo cliente, poderá ser necessário reprocessar os dados do canal para garantir que os dados ingeridos anteriormente estejam incluídos no conteúdo do conjunto de dados. Para obter mais informações, consulte Reprocessamento de dados do canal.

Formatos de arquivo

Atualmente, os datastores AWS IoT Analytics oferecem suporte aos formatos de arquivo JSON e Parquet. O formato de arquivo padrão é JSON.

- JSON (JavaScript Object Notation) — um formato de texto que suporta pares de nome-valor e listas ordenadas de valores.
- Apache Parquet — um formato de armazenamento colunar usado para armazenar e consultar com eficiência grandes volumes de dados.

Para configurar o formato de arquivo do datastore do AWS IoT Analytics, você pode usar o objeto `FileFormatConfiguration` ao criar o datastore.

`fileFormatConfiguration`

Contém as informações de configuração dos formatos de arquivo. Datastores do AWS IoT Analytics oferecem suporte a JSON e Parquet.

O formato de arquivo padrão é JSON. Você pode especificar apenas um formato. Não é possível alterar o formato do arquivo depois de criar o armazenamento de dados.

`jsonConfiguration`

Contém as informações de configuração do formato JSON.

`parquetConfiguration`

Contém as informações de configuração do formato Parquet.

`schemaDefinition`

Informações necessárias para definir um esquema.

`columns`

Especifica uma ou mais colunas que armazenam seus dados.

Cada esquema pode ter até 100 colunas. Cada coluna pode ter até 100 tipos aninhados.

`name`

O nome da coluna.

Restrições de comprimento: 1 a 255 caracteres.

`type`

O tipo de dados. Para obter mais informações sobre os tipos de dados compatíveis, consulte Tipos de dados comuns no Guia do desenvolvedor AWS Glue.

Restrições de comprimento: 1 a 131.072 caracteres.

AWS IoT Analytics suporta todos os tipos de dados listados na página Tipos de dados no Amazon Athena, exceto `DECIMAL(`*`precision`*`, `*`scale`*`)` - *`precision`*.

Criar um datastore (console)

O procedimento a seguir mostra como criar um datastore que salve dados no formato Parquet.

Para criar um datastore

1. Faça login em https://console.aws.amazon.com/iotanalytics/.

2. No painel de navegação, escolha Datastores.
3. Na página Datastores, selecione Criar datastore.
4. Na página Especificar detalhes do datastore, insira informações básicas sobre seu datastore.
   a. Em ID do datastore, insira uma ID exclusiva do datastore. Você não pode alterar a ID depois de criá-la.
   b. (Opcional) Em Tags, escolha Adicionar nova tag para adicionar uma ou mais tags personalizadas (pares chave-valor) ao seu datastore. As tags ajudam a identificar os recursos que você cria para AWS IoT Analytics.
   c. Escolha Next (Próximo).
5. Na página Configurar tipo de armazenamento, especifique como armazenar seus dados.
   a. Em Tipo de armazenamento, escolha Armazenamento gerenciado pelo serviço.
   b. Em Configurar quanto tempo você deseja manter seus dados processados, escolha Indefinidamente.
   c. Escolha Next (Próximo).
6. Na página Configurar formato de dados, defina a estrutura e o formato dos seus registros de dados.
   a. Para Classificação, escolha Parquet. Não é possível alterar o formato do arquivo depois de criar o datastore.
   b. Para a origem da inferência, escolha a string JSON para seu datastore.
   c. Em String, insira seu esquema no formato JSON, como no exemplo a seguir.

```
{
    "device_id": "0001",
    "temperature": 26,
    "humidity": 29,
    "datetime": "2018-01-26T07:06:01"
}
```

   d. Escolha Inferir esquema.
   e. Em Configurar esquema do Parquet, confirme se o formato corresponde ao seu exemplo JSON. Se o formato não corresponder, atualize o esquema do Parquet manualmente.
      - Se você quiser que seu esquema mostre mais colunas, escolha Adicionar nova coluna, insira o nome da coluna e escolha o tipo de dados.

> **Note**
>
> Por padrão, você pode ter 100 colunas para seu esquema. Para obter mais informações, consulte as AWS IoT Analytics cotas.

- Você pode alterar o tipo de dados de uma coluna existente. Para obter mais informações sobre os tipos de dados compatíveis, consulte Tipos de dados comuns no Guia do desenvolvedor AWS Glue.

  > **Note**
  >
  > Depois que você criar seu datastore, não será possível alterar o tipo de dados de uma coluna existente.

- Para remover uma coluna existente, escolha Remover coluna.

f. Escolha Next (Próximo).

7. (Opcional) AWS IoT Analytics oferece suporte a partições personalizadas em seu datastore para que você possa consultar dados removidos para melhorar a latência. Para obter mais informações sobre partições personalizadas compatíveis, consulte Partições personalizadas.

   Escolha Next (Próximo).

8. Na página Revisar e criar, revise suas escolhas e, em seguida, selecione Criar datastore,

   > **Important**
   >
   > Não é possível alterar a ID do datastore, o formato do arquivo ou o tipo de dados de uma coluna depois que você criar o datastore.

9. Verifique se seu novo datastore aparece na página Datastores.

Partições personalizadas

AWS IoT Analytics oferece suporte ao particionamento de dados para que você possa organizar os dados em seu datastore. Ao usar o particionamento de dados para organizar dados, você pode consultar dados eliminados. Isso diminui a quantidade de dados examinados por consulta e melhora a latência.

Você pode particionar seus dados de acordo com os atributos dos dados da mensagem ou os atributos adicionados por meio das atividades do pipeline.

Para começar, habilite o particionamento de dados em um datastore. Especifique uma ou mais dimensões de partição de dados e conecte seu datastore particionado a um pipeline AWS IoT Analytics. Em seguida, escreva consultas que aproveitem a cláusula `WHERE` para otimizar o desempenho.

Criar um datastore (console)

O procedimento a seguir mostra como criar um datastore com uma partição personalizada.

Para criar um datastore

1. Faça login no console do AWS IoT Analytics.
2. No painel de navegação, escolha Datastores.
3. Na página Datastores, selecione Criar datastore.
4. Na página Especificar detalhes do datastore, insira informações básicas sobre seu datastore.
   a. Em ID do datastore, insira uma ID exclusiva do datastore. Você não pode alterar a ID depois de criá-la.
   b. (Opcional) Em Tags, escolha Adicionar nova tag para adicionar uma ou mais tags personalizadas (pares chave-valor) ao seu datastore. As tags ajudam a identificar os recursos para os quais você cria AWS IoT Analytics.
   c. Escolha Next (Próximo).
5. Na página Configurar tipo de armazenamento, especifique como armazenar seus dados.
   a. Em Tipo de armazenamento, escolha Armazenamento gerenciado pelo serviço.
   b. Em Configurar quanto tempo você deseja manter seus dados processados, escolha Indefinidamente.
   c. Escolha Next (Próximo).
6. Na página Configurar formato de dados, defina a estrutura e o formato dos seus registros de dados.
   a. Para a Classificação do formato de dados do seu datastore, escolha JSON ou Parquet. Para obter mais informações sobre os tipos de arquivo compatíveis AWS IoT Analytics, consulte Formatos de arquivo.

> **Note**
>
> Não é possível alterar o formato do arquivo depois de criar o datastore.

b. Escolha Next (Próximo).

7. Crie partições personalizadas para esse datastore.

   a. Em Adicionar partições de dados, selecione Ativar.

   b. Em Origem da partição de dados, especifique as informações básicas sobre a origem da sua partição.

      Escolha Origem da amostra e selecione o canal AWS IoT Analytics que coleta mensagens para esse datastore.

   c. Em Atributos de amostra de mensagem, selecione os atributos de mensagem que você deseja usar para particionar seu datastore. Em seguida, adicione suas seleções como dimensões de partição de atributo ou dimensões de partição com timestamp em Ações.

      > **Note**
      >
      > Você pode adicionar somente uma partição de timestamp ao seu datastore.

   d. Para Dimensões personalizadas da partição do datastore, defina as informações básicas sobre as dimensões da partição. Cada atributo de amostra de mensagem que você selecionou na etapa anterior se tornará as dimensões da sua partição. Personalize cada dimensão com estas opções:

      - Tipo de partição — especifique se essa dimensão de partição é um Atributo ou um tipo de partição Timestamp.
      - Nome do atributo e Nome da dimensão — por padrão, AWS IoT Analytics usará o nome do atributo de amostra de mensagem que você selecionou como um identificador para a dimensão da partição do atributo. Edite o nome do atributo para personalizar o nome da dimensão da partição. Você pode usar o nome da dimensão na cláusula `WHERE` para otimizar o desempenho da consulta.
        - O nome de qualquer dimensão de atributo de partição é prefixado com `__partition_`.

- Para tipos de partição com timestamp, AWS IoT Analytics cria as quatro dimensões a seguir com nomes `__year`, `__month`, `__day`, `__hour`.
- Ordenação — reorganize as dimensões da partição para melhorar a latência de suas consultas.

Para o formato Timestamp, especifique o formato da partição de timestamp combinando o timestamp ingerido dos dados da mensagem. Você pode escolher uma das opções de formato AWS IoT Analytics listadas ou especificar uma que corresponda ao formato dos seus dados. Saiba mais sobre como especificar formatadores de data e hora.

Para adicionar uma nova dimensão que não seja um atributo de mensagem, escolha Adicionar novas partições.

e. Escolha Next (Próximo).

8. Na página Revisar e criar, revise suas escolhas e, em seguida, selecione Criar datastore.

> **Important**
>
> - Não é possível alterar a ID do datastore depois de criar o datastore.
> - Para editar partições existentes, você deve criar outro datastore e reprocessar os dados por meio de um pipeline.

9. Verifique se seu novo datastore aparece na página Datastores.

Criando um pipeline

Um pipeline consome as mensagens de um canal e permite processar e filtrar as mensagens antes de armazená-las em um datastore. Para conectar um canal a um datastore, crie um pipeline. O pipeline mais simples possível não contém atividades que não sejam especificar o canal que coleta os dados e identificar o datastore para o qual as mensagens são enviadas. Para obter mais informações sobre pipelines mais complicados, consulte Atividades do pipeline.

Ao iniciar, recomendamos criar um pipeline que não faça nada além de conectar um canal a um datastore. Depois, após verificar os fluxos de dados brutos ao datastore, é possível introduzir atividades adicionais do pipeline para processas esses dados.

Execute o comando a seguir para criar um pipeline.

```
aws iotanalytics create-pipeline --cli-input-json file://mypipeline.json
```

O arquivo `mypipeline.json` contém o conteúdo a seguir.

```
{
    "pipelineName": "mypipeline",
    "pipelineActivities": [
        {
            "channel": {
                "name": "mychannelactivity",
                "channelName": "mychannel",
                "next": "mystoreactivity"
            }
        },
        {
            "datastore": {
                "name": "mystoreactivity",
                "datastoreName": "mydatastore"
            }
        }
    ]
}
```

Execute o comando a seguir para listar os pipelines existentes.

```
aws iotanalytics list-pipelines
```

Execute o comando a seguir para visualizar a configuração de um pipeline individual.

```
aws iotanalytics describe-pipeline  --pipeline-name mypipeline
```

Consumir dados para o AWS IoT Analytics

Se você tem um canal que roteia dados pra um pipeline que armazena os dados em um datastore onde eles podem ser consultados, está pronto para enviar dados de mensagem para o AWS IoT Analytics. Veja a seguir dois métodos para inserir dados no AWS IoT Analytics. Você pode enviar uma mensagem usando o agente de mensagens AWS IoT ou usar a `BatchPutMessage` da API do AWS IoT Analytics.

Tópicos

- Usando o agente de mensagens AWS IoT
- Uso da API BatchPutMessage

Usando o agente de mensagens AWS IoT

Para usar o agente de mensagens do AWS IoT, crie uma regra usando o Rules Engine do AWS IoT. A regra roteia mensagens com um tópico específico para o AWS IoT Analytics. No entanto, essa regra exige que primeiro você crie uma função que conceda as permissões necessárias.

Criar um perfil do IAM

Para que as mensagens do AWS IoT sejam encaminhadas para um canal do AWS IoT Analytics, configure uma regra. No entanto, antes é necessário criar um perfil do IAM que conceda permissão a essa regra para enviar dados da mensagem a um canal do AWS IoT Analytics.

Execute o comando da a seguir para criar a função.

```
aws iam create-role --role-name myAnalyticsRole --assume-role-policy-document file://
arpd.json
```

O conteúdo do arquivo `arpd.json` deve ser semelhante ao seguinte exemplo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iot.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Em seguida, anexe um documento de política para a função.

```
aws iam put-role-policy --role-name myAnalyticsRole --policy-name myAnalyticsPolicy --
policy-document file://pd.json
```

O conteúdo do arquivo `pd.json` deve ser semelhante ao seguinte exemplo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotanalytics:BatchPutMessage",
            "Resource": [
               "arn:aws:iotanalytics:us-west-2:your-account-number:channel/mychannel"
            ]
        }
    ]
}
```

Criando uma Regra AWS IoT

Crie uma regra para o AWS IoT que envia mensagens ao canal.

```
aws iot create-topic-rule --rule-name analyticsTestRule --topic-rule-payload file://
rule.json
```

O conteúdo do arquivo `rule.json` deve ser semelhante ao seguinte exemplo:

```
{
    "sql": "SELECT * FROM 'iot/test'",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [ {
        "iotAnalytics": {
            "channelName": "mychannel",
            "roleArn": "arn:aws:iam::your-account-number:role/myAnalyticsRole"
        }
    } ]
}
```

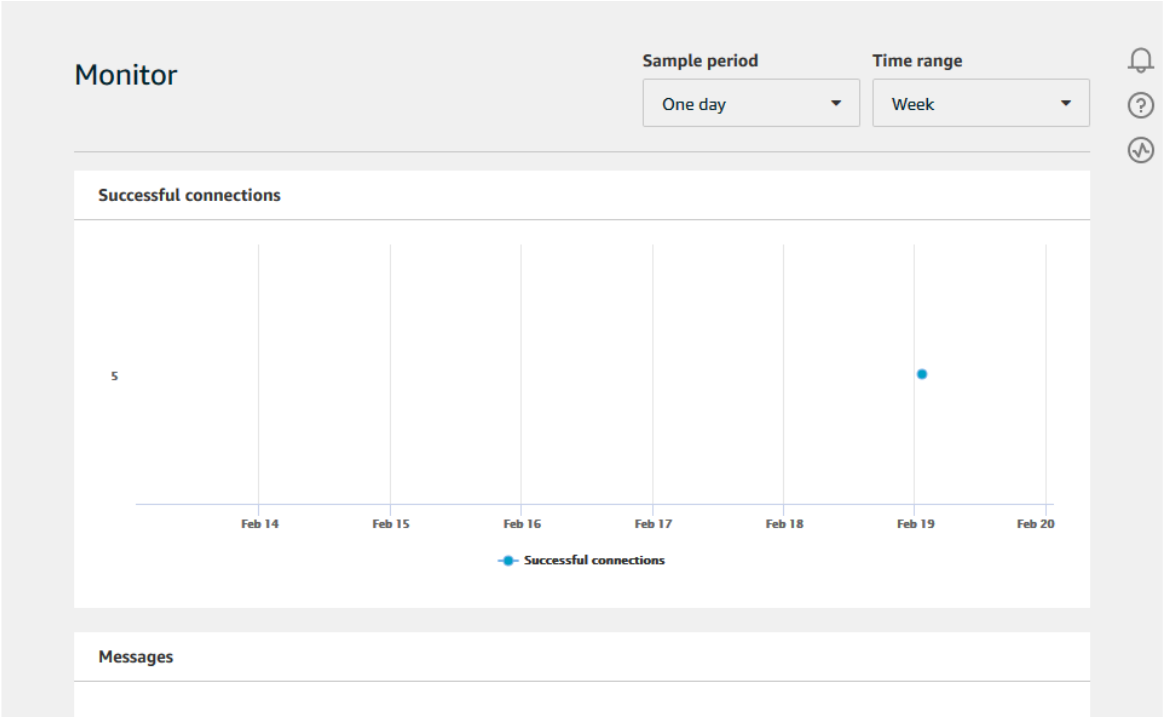
Substitua `iot/test` pelo tópico MQTT das mensagens que devem ser roteadas. Substitua o nome do canal e a função por aqueles criados nas seções anteriores.

Enviando mensagens MQTT para AWS IoT Analytics

Depois de ter acrescentado uma regra a um canal, um canal a um pipeline e um pipeline a um datastore, qualquer dado que corresponda à regra fluirá pelo AWS IoT Analytics até o datastore pronto para ser consultado. Para testar isso, é possível usar o console do AWS IoT para enviar uma mensagem.

> **Note**
>
> Os nomes de campo de cargas de mensagem (dados) que você envia ao AWS IoT Analytics:
>
> - Devem conter apenas caracteres alfanuméricos e sublinhados (_). Outros caracteres especiais não são permitidos.
> - Devem começar com um caractere alfabético ou com um sublinhado (_).
> - Não podem conter hífens (-).
> - Em termos de expressões regulares: "`^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$`".
> - Não podem ser maiores que 255 caracteres.
> - Não diferenciam maiúsculas de minúsculas. Campos denominados `foo` e `F00` na mesma carga útil são considerados duplicatas.
>
> Por exemplo, `{"temp_01": 29}` ou `{"_temp_01": 29}` são válidos, mas `{"temp-01": 29}`, `{"01_temp": 29}` ou `{"__temp_01": 29}` são inválidos em cargas de mensagem.

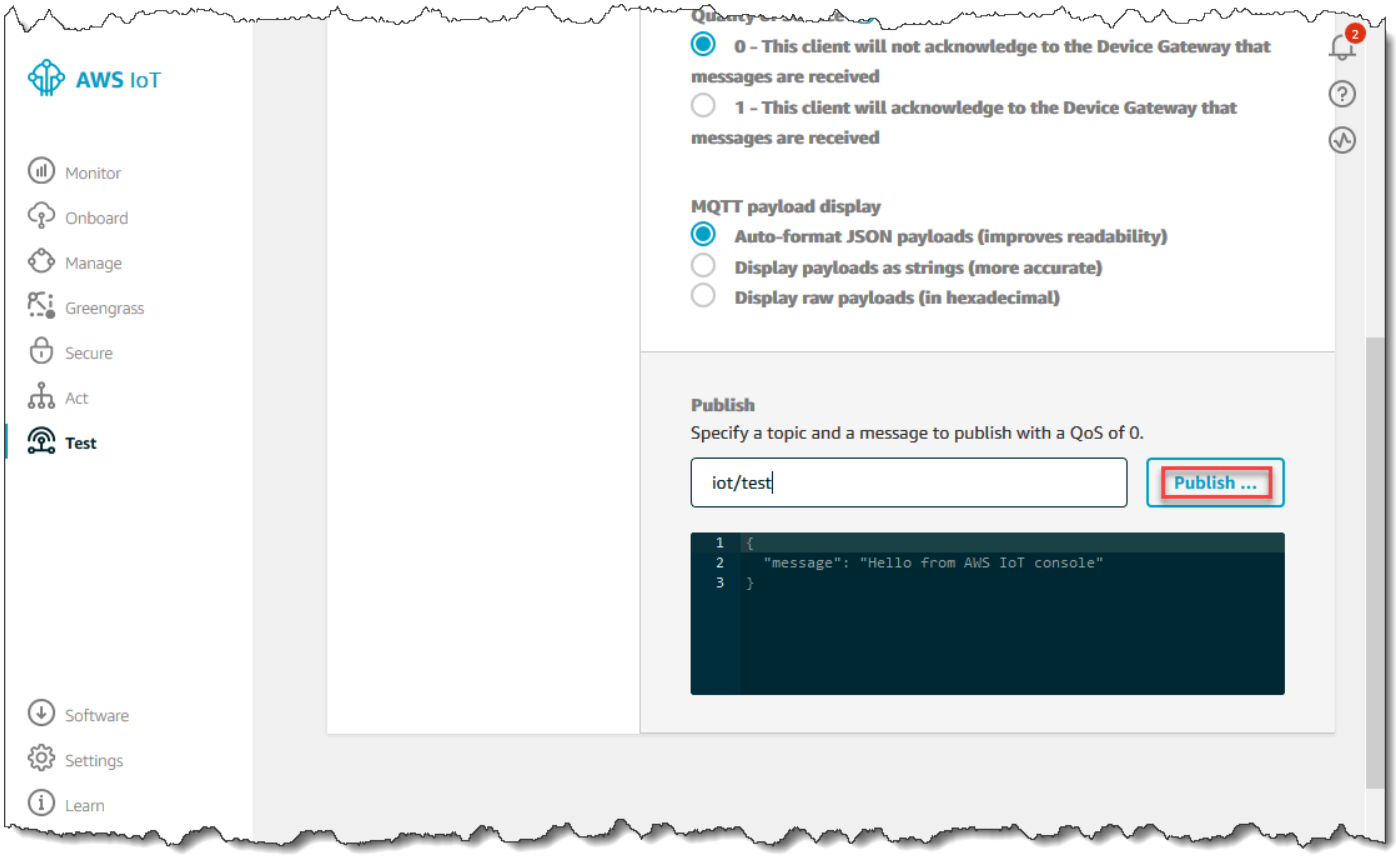
1. No console da AWS IoT, no painel de navegação à esquerda, selecione Ação.



2. Na página MQTT do cliente, na seção Publicar, em Especificar um tópico, digite **`iot/test`**. Na seção de carga útil da mensagem, verifique se o conteúdo do JSON está presente ou digite-o se não estiver.

```
{
    "message": "Hello from the IoT console"
}
```

3. Selecione Publicar em um tópico.



Isso publica uma mensagem que é roteado para o datastore que você criou anteriormente.

Uso da API BatchPutMessage

Outra forma de inserir dados de mensagens ao AWS IoT Analytics é usar o comando da API `BatchPutMessage`. Esse método não exige que você configure uma regra de AWS IoT para rotear mensagens com um tópico específico a seu canal. Mas exige que o dispositivo que envia seus dados/mensagens ao canal possa executar software criado com o SDK da AWS ou possa usar a AWS CLI para chamar `BatchPutMessage`.

1. Crie um arquivo `messages.json` contendo as mensagens a serem enviadas (neste exemplo, apenas uma mensagem é enviada).

```
[
    { "messageId": "message01", "payload": "{ \"message\": \"Hello from the CLI
\" }" }
]
```

2. Execute o comando `batch-put-message`.

```
aws iotanalytics batch-put-message  --channel-name mychannel  --messages file://
messages.json --cli-binary-format raw-in-base64-out
```

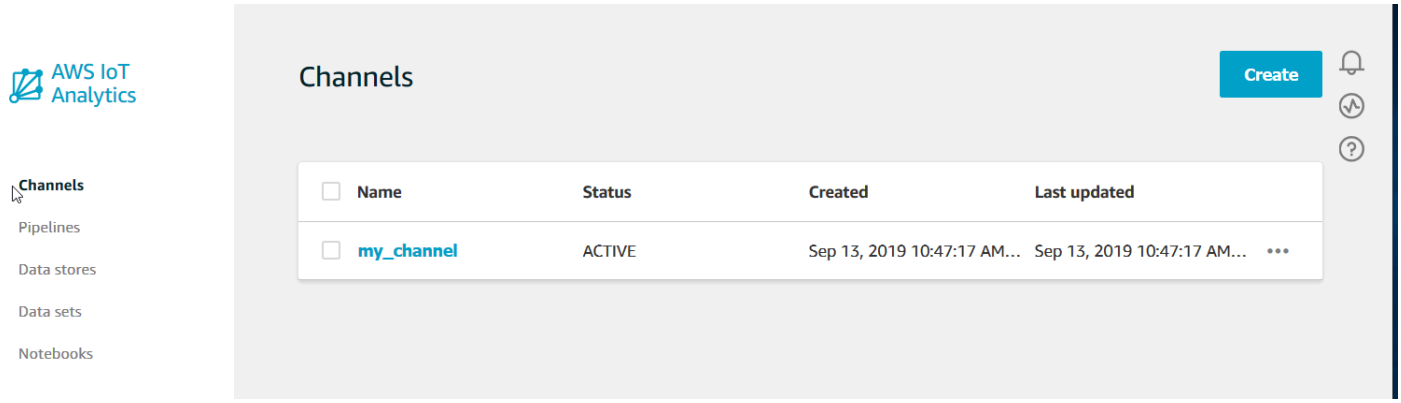
Se não houver erros, você verá a saída a seguir.

```
{
    "batchPutMessageErrorEntries": []
}
```

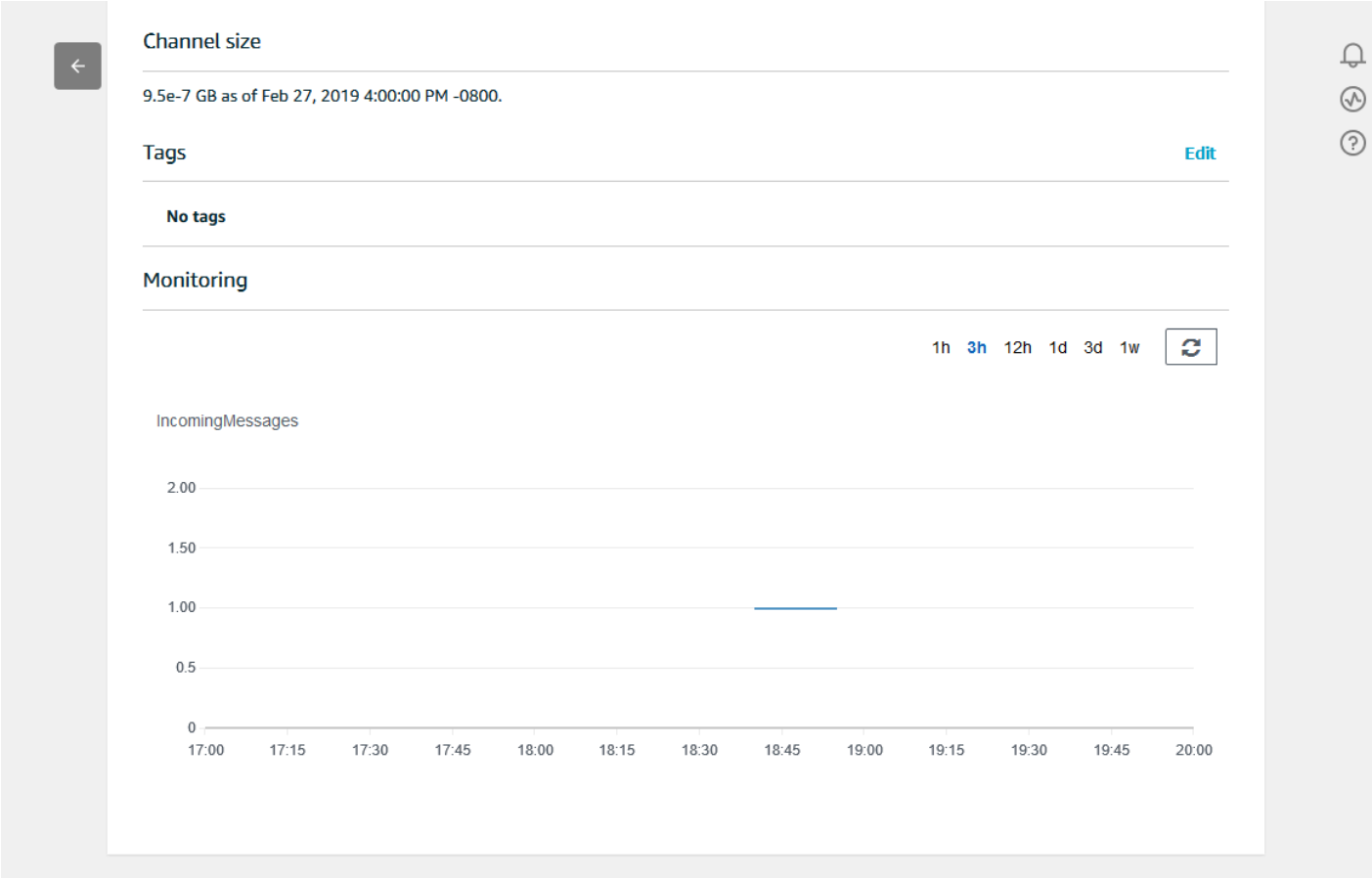
Monitorando os dados ingeridos

É possível verificar se as mensagens enviadas estão sendo ingeridas no canal usando o console do AWS IoT Analytics.

1. No console do AWS IoT Analytics, no painel de navegação à esquerda, selecione Preparar e (se necessário) selecione Canais e escolha o nome do canal criado anteriormente.

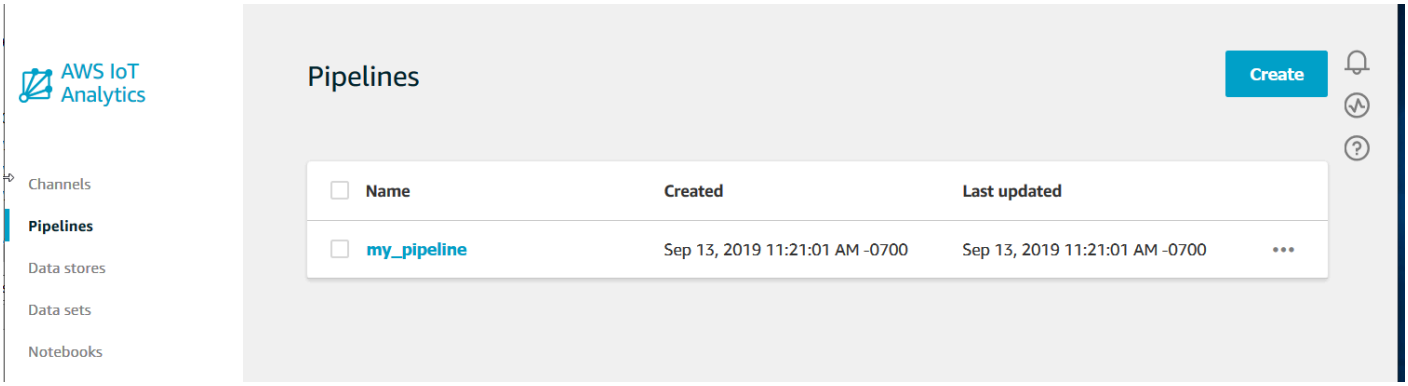


2. Na página de detalhes do canal, role até a seção Monitoring (Monitoramento). Ajuste o período exibido conforme necessário escolhendo um dos indicadores de período (1h 3h 12h 1d 3d 1w (1h 3h 12h 1d 3d 1s)). Deve ser exibida uma linha de gráfico indicando o número de mensagens ingeridas nesse canal durante o período especificado:



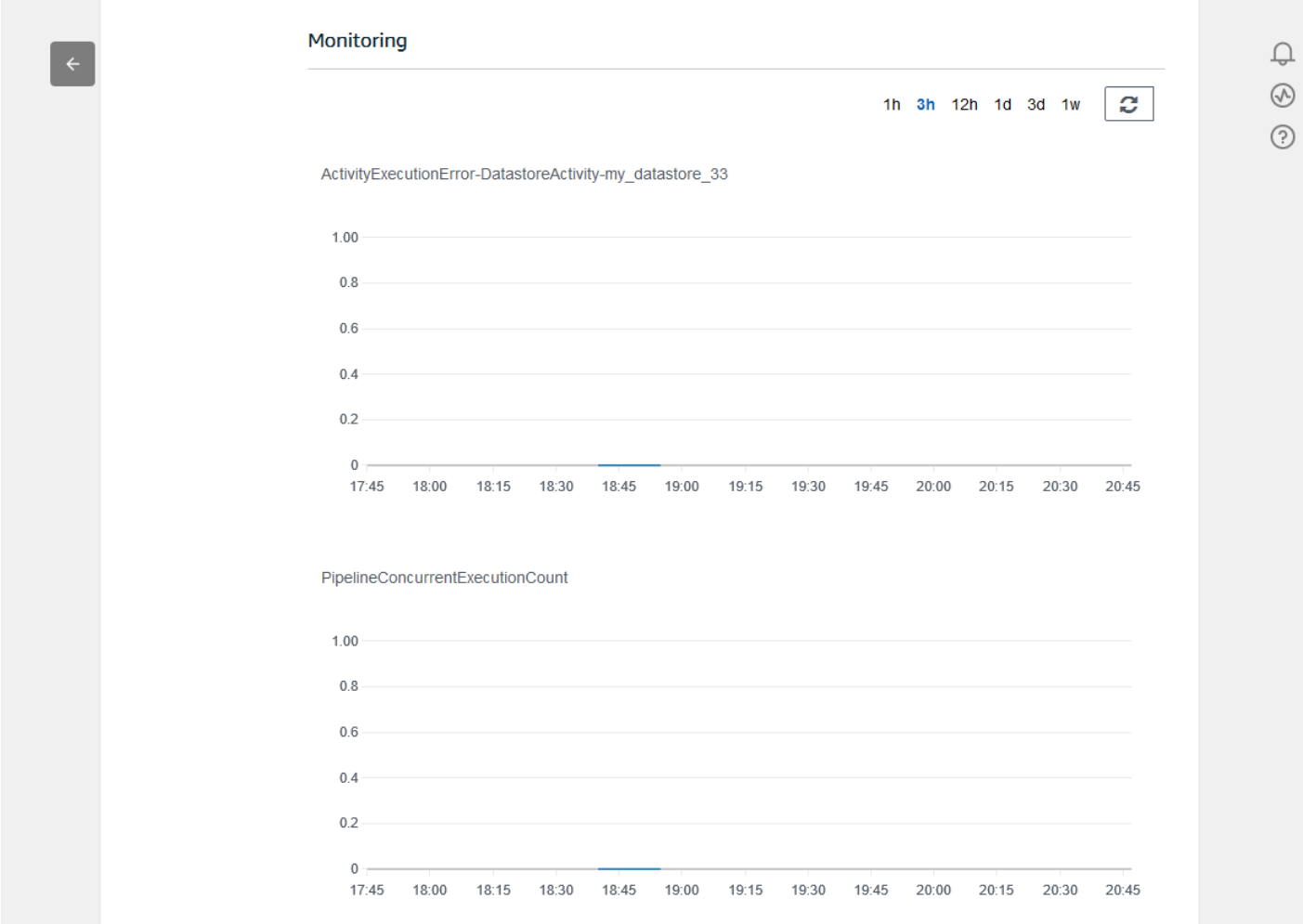
Um recurso de monitoramento semelhante existe para verificar execuções de atividades do pipeline. É possível monitorar erros de execução de atividade na página de detalhes do pipeline. Se você ainda não tiver especificado atividades como parte do pipeline, serão exibidos 0 erros de execução.

1. No console do AWS IoT Analytics, no painel de navegação à esquerda, selecione Preparar, selecione Pipelines e escolha o nome de um pipeline criado anteriormente.



2. Na página de detalhes do pipeline, role até a seção Monitoring (Monitoramento). Ajuste o período exibido conforme necessário escolhendo um dos indicadores de período (1h 3h 12h 1d 3d 1w

(1h 3h 12h 1d 3d 1s)). Deve ser exibida uma linha de gráfico indicando o número de erros de execução de atividades do pipeline durante o período especificado.



Criação de um conjunto de dados

Você recupera dados de um armazenamento de dados criando um conjunto de dados SQL ou um conjunto de dados de contêiner. AWS IoT Analytics pode consultar os dados para responder a perguntas analíticas. Embora um datastore não seja um banco de dados, você pode usar expressões SQL para consultar os dados e produzir resultados que estão armazenados em um conjunto de dados.

Tópicos

- Consultar dados
- Acessando os dados consultados

Consultar dados

Para consultar os dados, crie um conjunto de dados. Um conjunto de dados contém o SQL usado para consultar o datastore juntamente com um agendamento adicional que repete a consulta em um dia e horário de sua escolha. Você cria programações opcionais usando expressões semelhantes às expressões de programação do Amazon CloudWatch.

Execute o comando a seguir para criar um conjunto de dados.

```
aws iotanalytics create-dataset --cli-input-json file://mydataset.json
```

Onde o arquivo `mydataset.json` contém o seguinte conteúdo:

```
{
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName":"myaction",
            "queryAction": {
                "sqlQuery": "select * from mydatastore"
            }
        }
    ]
}
```

Execute o comando a seguir para criar o conteúdo do conjunto de dados executando a consulta.

```
aws iotanalytics create-dataset-content --dataset-name mydataset
```

Aguarde alguns minutos para que o conteúdo do conjunto de dados seja criado antes de continuar.

Acessando os dados consultados

O resultado da consulta é o conteúdo do conjunto de dados, armazenado como um arquivo no formato CSV. O arquivo é disponibilizado por meio do Amazon S3. O exemplo a seguir mostra como você pode verificar se os resultados estão prontos e fazer download do arquivo.

Execute o seguinte comando `get-dataset-content`.

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

Se o conjunto de dados contiver dados, a saída do `get-dataset-content` terá `"state": "SUCCEEDED"` no campo `status`, como o seguinte exemplo:

```
{
    "timestamp": 1508189965.746,
    "entries": [
        {
          "entryName": "someEntry",
          "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-Amz-"

        }
    ],
    "status": {
      "state": "SUCCEEDED",
      "reason": "A useful comment."
    }
}
```

`dataURI` é uma URL assinada para os resultados de saída. Tem validade por um curto período de tempo (algumas horas). Dependendo do seu fluxo de trabalho, você sempre pode chamar `get-dataset-content` antes de acessar o conteúdo, porque chamar esse comando gera uma nova URL assinada.

Explorar dados AWS IoT Analytics

Você tem diversas opções para armazenar, analisar e visualizar seus dados AWS IoT Analytics.

Tópicos nesta página:

- Amazon S3
- AWS IoT Events
- Amazon QuickSight
- Bloco de anotações Jupyter

Amazon S3

É possível enviar conteúdo do conjunto de dados para um bucket do Amazon Simple Storage Service (Amazon S3), permitindo a integração com os data lakes existentes ou

o acesso usando aplicativos internos e ferramentas de visualização. Consulte o campo `contentDeliveryRules::destination::s3DestinationConfiguration` em CreateDataset.

AWS IoT Events

É possível enviar conteúdo do conjunto de dados como uma entrada para o AWS IoT Events, um serviço que permite monitorar dispositivos ou processos em busca de falhas ou alterações na operação, além de acionar ações adicionais quando esses eventos ocorrerem.

Para fazer isso, crie um conjunto de dados usando CreateDataset e especifique uma entrada AWS IoT Events no campo `contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. Você também deve especificar o `roleArn` da função que concede permissão do AWS IoT Analytics para executar "iotevents:BatchPutMessage". Sempre que o conteúdo do conjunto de dados for criado, AWS IoT Analytics enviará cada entrada de conteúdo do conjunto de dados como uma mensagem para a entrada AWS IoT Events especificada. Por exemplo, se o seu conjunto de dados contém:

```
"what","who","dt"
"overflow","sensor01","2019-09-16 09:04:00.000"
"overflow","sensor02","2019-09-16 09:07:00.000"
"underflow","sensor01","2019-09-16 11:09:00.000"
...
```

em seguida, AWS IoT Analytics enviará mensagens contendo campos como este:

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

e você desejará criar uma entrada AWS IoT Events que reconheça os campos nos quais você está interessado (um ou mais dos `what`, `who`, `dt`) e criar um modelo de AWS IoT Events detector que use esses campos de entrada em eventos para acionar ações ou definir variáveis internas.

Amazon QuickSight

O AWS IoT Analytics fornece integração direta com o Amazon QuickSight. O Amazon QuickSight é um serviço de análise de negócios rápido que você pode usar para criar visualizações, realizar uma análise ad hoc e obter insights de seus dados rapidamente. O Amazon QuickSight permite que as

organizações escalem para centenas de milhares de usuários e fornece desempenho responsivo usando um mecanismo de memória robusto (SPICE). O Amazon QuickSight está disponível nestas regiões.

Bloco de anotações Jupyter

Os conjuntos de dados do AWS IoT Analytics também podem ser consumidos diretamente pelo caderno Jupyter para realizar análises avançadas e exploração de dados. O caderno Jupyter é uma solução de código aberto. Você pode instalar e fazer download em http://jupyter.org/install.html. A integração adicional com o SageMaker, uma solução de blocos de anotações hospedada pela Amazon também está disponível.

Mantendo várias versões dos conjuntos de dados

É possível escolher quantas versões do conteúdo do conjunto de dados devem ser retidas e por quanto tempo especificando valores para os campos `retentionPeriod and versioningConfiguration` do conjunto de dados ao invocar as APIs CreateDataset e UpdateDataset:

```
...
"retentionPeriod": {
  "unlimited": "boolean",
  "numberOfDays": "integer"
},
"versioningConfiguration": {
  "unlimited": "boolean",
  "maxVersions": "integer"
},
...
```

As configurações desses dois parâmetros funcionam em conjunto para determinar quantas versões do conteúdo do conjunto de dados são retidas e por quanto tempo das seguintes maneiras:

	retentionPeriod	retentionPeriod:	retentionPeriod:
	[não especificado]	unlimited = TRUE, numberOfDays = não definido	ilimitado = FALSE, numberOfDays = X

versioningConfiguration: [não especificado]	Somente a versão mais recente e a última versão bem-sucedida (se for diferente) são retidas por 90 dias.	Somente a versão mais recente e a última versão bem-sucedida (se for diferente) são retidas por um tempo ilimitado.	Somente a versão mais recente e a última versão bem-sucedida (se for diferente) são retidas por X dias.
versioningConfiguration: unlimited = TRUE, maxVersions não definido	Todas as versões dos últimos 90 dias serão retidas, independentemente de quantas.	Não há limite para o número de versões retidas.	Todas as versões dos últimos X dias serão retidas, independentemente de quantas.
versioningConfiguration: unlimited = FALSE, maxVersions = Y	No máximo Y versões dos últimos 90 dias serão retidas.	Até Y versões serão retidas, independentemente do tempo de existência.	No máximo Y versões dos últimos X dias serão retidas.

Sintaxe da carga útil da mensagem

Os nomes de campo de cargas úteis de mensagem (dados) que você envia ao AWS IoT Analytics:

- Devem conter apenas caracteres alfanuméricos e sublinhados (_); outros caracteres especiais não são permitidos.
- Devem começar com um caractere alfabético ou com um sublinhado (_).
- Não podem conter hifens (-).
- Em termos de expressões regulares: “`^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$`”.
- Não podem ser maiores que 255 caracteres.
- Não diferenciam maiúsculas de minúsculas. Campos denominados “foo” e “FOO” na mesma carga útil são considerados duplicatas.

Por exemplo, `{"temp_01": 29}` ou `{"_temp_01": 29}` são válidos, mas `{"temp-01": 29}`, `{"01_temp": 29}` ou `{"__temp_01": 29}` são inválidos em cargas úteis de mensagem.

Trabalho com dados AWS IoT SiteWise

O AWS IoT SiteWise é um serviço gerenciado que permite coletar, modelar, analisar e visualizar dados de equipamentos industriais em escala. O serviço fornece uma estrutura de modelagem de ativos que pode ser usada para criar representações de seus dispositivos industriais, processos e instalações.

Com modelos de ativos AWS IoT SiteWise, você define quais dados de equipamentos industriais devem ser consumidos e como processar seus dados em métricas complexas. Você pode configurar modelos de ativos para coletar e processar dados na Nuvem AWS. Para obter mais informações, consulte o Manual do usuário da AWS IoT SiteWise.

AWS IoT Analytics se integra a AWS IoT SiteWise para que você possa executar e programar consultas SQL nos dados AWS IoT SiteWise. Para começar a consultar seus dados AWS IoT SiteWise, crie um armazenamento de dados seguindo os procedimentos de Definir configurações de armazenamento no Guia do usuário AWS IoT SiteWise. Em seguida, siga as etapas inseridas no Crie um conjunto de dados com dados AWS IoT SiteWise (console) ou em Criar um conjunto de dados com dados AWS IoT SiteWise (AWS CLI) para criar um conjunto de dados AWS IoT Analytics e executar uma consulta SQL em seus dados industriais.

Tópicos

- Crie um conjunto de dados AWS IoT Analytics com dados AWS IoT SiteWise
- Acessar o conteúdo do conjunto de dados
- Tutorial: consultar AWS IoT SiteWise dados em AWS IoT Analytics

Crie um conjunto de dados AWS IoT Analytics com dados AWS IoT SiteWise

Um conjunto de dados AWS IoT Analytics contém instruções e expressões SQL usadas para consultar o datastore juntamente com uma programação adicional que repete a consulta em um dia e horário que você especifica. Você pode usar expressões semelhantes às Expressões de programação do Amazon CloudWatch para criar programações opcionais.

> **Note**
>
> Um conjunto de dados geralmente é uma coleção de dados que podem ou não estar organizados em formato tabular. Por outro lado, AWS IoT Analytics cria seu conjunto de dados aplicando uma consulta SQL aos dados em seu datastore.

Siga essas etapas para começar a criar um conjunto de dados para seus dados AWS IoT SiteWise.

Tópicos

- Crie um conjunto de dados com dados AWS IoT SiteWise (console)
- Criar um conjunto de dados com dados AWS IoT SiteWise (AWS CLI)

Crie um conjunto de dados com dados AWS IoT SiteWise (console)

Use essas etapas para criar um conjunto de dados no console AWS IoT Analytics para seus dados AWS IoT SiteWise.

Para criar um conjunto de dados

1. Em https://console.aws.amazon.com/iotanalytics/, no painel de navegação esquerdo, escolha Conjuntos de dados.
2. Na página Criar conjunto de dados, escolha Criar SQL.
3. Na página Especificar detalhes do conjunto de dados, especifique os detalhes do seu conjunto de dados.

   a. Digite um nome para o conjunto de dados.

   b. Em Fonte do datastore, escolha a ID exclusiva que identifica seu datastore AWS IoT SiteWise.

   c. (Opcional) Em Tags, adicione uma ou mais tags personalizadas (pares chave-valor) ao seu conjunto de dados.
4. Use expressões SQL para consultar seus dados e responder perguntas analíticas.

   a. No campo Consulta do autor, insira uma consulta SQL que usa um curinga para exibir até cinco linhas de dados.

```
SELECT * FROM my_iotsitewise_datastore.asset_metadata LIMIT 5
```

Para receber mais informações sobre a funcionalidade SQL compatível com AWS IoT Analytics, consulte Expressões SQL em AWS IoT Analytics. Ou veja Tutorial: consultar AWS IoT SiteWise dados em AWS IoT Analytics para exemplos de consultas estatísticas que podem apresentar informações sobre seus dados.

b. Você pode escolher Consulta de teste para validar se sua entrada está correta e exibir os resultados em uma tabela após a consulta.

> **Note**
>
> Como Amazon Athena limita o número máximo de consultas em execução, você deve limitar sua consulta SQL a um tamanho razoável para que ela não seja executada por um período prolongado.

5. (Opcional) Quando você cria o conteúdo do conjunto de dados usando dados de um período especificado, alguns dados podem não chegar a tempo de serem processados. Para permitir um atraso, você pode especificar um deslocamento ou delta. Para obter mais informações, consulte Receber notificações de dados atrasadas por meio do Amazon CloudWatch Events.

   Depois de configurar um filtro de seleção de dados na página Configurar filtro de seleção de dados, escolha Próximo.

6. (Opcional) Na página Definir programação de consulta, você pode programar essa consulta para ser executada regularmente para atualizar o conjunto de dados. As programações de conjuntos de dados podem ser criadas e editadas a qualquer momento.

   > **Note**
   >
   > Dados de ingestões de AWS IoT SiteWise a AWS IoT Analytics cada seis horas. Recomendamos selecionar uma frequência de seis horas ou mais.

   Escolha uma opção para Frequência e, depois, escolha Próximo.

7. AWS IoT Analytics criará versões desse conteúdo do conjunto de dados e armazenará seus resultados de análise pelo período especificado. Recomendamos 90 dias, mas você pode optar por definir sua política de retenção personalizada. Você também pode limitar o número de versões armazenadas do conteúdo do seu conjunto de dados.

Depois de selecionar suas opções na página Configurar os resultados do seu conjunto de dados, escolha Próximo.

8. (Opcional) Você pode configurar as regras de entrega dos resultados do seu conjunto de dados para um destino específico, como AWS IoT Events.

   Depois de selecionar suas opções na página Configurar regras de entrega de conteúdo do conjunto de dados, escolha Próximo.

9. Verifique suas escolhas e selecione Criar conjunto de dados.
10. Verifique se seu novo conjunto de dados aparece na página Conjuntos de dados.

Criar um conjunto de dados com dados AWS IoT SiteWise (AWS CLI)

Execute os comandos AWS CLI a seguir para começar a consultar seus dados AWS IoT SiteWise.

Os exemplos mostrados aqui usam o AWS Command Line Interface (AWS CLI). Para obter mais informações em AWS CLI, consulte o Guia do usuário AWS Command Line Interface. Para obter mais informações sobre os comandos da CLI disponíveis para AWS IoT Analytics, consulte iotanalytics na Referência AWS Command Line Interface.

Para criar um conjunto de dados

1. Execute o comando `create-dataset` a seguir para criar um conjunto de dados.

   ```
   aws iotanalytics create-dataset --cli-input-json file://my_dataset.json
   ```

   Onde o arquivo `my_dataset.json` contém o seguinte conteúdo:

   ```
   {
       "datasetName": "my_dataset",
       "actions": [
           {
               "actionName":"my_action",
               "queryAction": {
                   "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
    LIMIT 5"
               }
           }
       ]
   ```

```
}
```

Para receber mais informações sobre a funcionalidade SQL compatível com AWS IoT Analytics, consulte Expressões SQL em AWS IoT Analytics. Ou veja Tutorial: consultar AWS IoT SiteWise dados em AWS IoT Analytics para exemplos de consultas estatísticas que podem apresentar informações sobre seus dados.

2. Execute o comando `create-dataset-content` a seguir para criar o conteúdo do conjunto de dados executando sua consulta.

```
aws iotanalytics create-dataset-content --dataset-name my_dataset
```

Acessar o conteúdo do conjunto de dados

O resultado da consulta SQL é o conteúdo do conjunto de dados, armazenado como um arquivo no formato CSV. O arquivo é disponibilizado por meio do Amazon S3. O exemplo a seguir mostra como você pode verificar se os resultados estão prontos e fazer download do arquivo.

Tópicos

- Acessar o conteúdo do conjunto de dados no AWS IoT Analytics (Console)
- Acessar o conteúdo do conjunto de dados em AWS IoT Analytics (AWS CLI)

Acessar o conteúdo do conjunto de dados no AWS IoT Analytics (Console)

Se seu conjunto de dados contiver algum dado, você poderá visualizar e baixar os resultados da consulta SQL no console AWS IoT Analytics.

Para acessar os resultados do seu conjunto de dados AWS IoT Analytics

1. No console, na página Conjuntos de dados, escolha o nome do conjunto de dados que você deseja acessar.
2. Na página de resumo do conjunto de dados, escolha a guia Conteúdo.
3. Na tabela Conteúdo do conjunto de dados, escolha o nome da consulta na qual você deseja visualizar os resultados ou faça o download de um arquivo csv dos resultados.

Acessar o conteúdo do conjunto de dados em AWS IoT Analytics (AWS CLI)

Se seu conjunto de dados contiver algum dado, você poderá visualizar e baixar os resultados da consulta SQL.

Os exemplos mostrados aqui usam o AWS Command Line Interface (AWS CLI). Para obter mais informações em AWS CLI, consulte o Guia do usuário AWS Command Line Interface. Para obter mais informações sobre os comandos da CLI disponíveis para o AWS IoT Analytics, consulte iotanalytics na Referência AWS Command Line Interface.

Para acessar os resultados do seu conjunto de dados do AWS IoT Analytics (AWS CLI)

1. Execute o comando `get-dataset-content` a seguir para ver o resultado da sua consulta.

```
aws iotanalytics get-dataset-content --dataset-name my_iotsitewise_dataset
```

2. Se seu conjunto de dados contiver algum dado, então a saída de `get-dataset-content` tem `"state": "SUCCEEDED"` no campo `status`, como no exemplo a seguir.

```
{
    "timestamp": 1508189965.746,
    "entries": [
        {
          "entryName": "my_entry_name",
          "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-
Amz-"

        }
    ],
    "status": {
      "state": "SUCCEEDED",
      "reason": "A useful comment."
    }
}
```

3. A saída de `get-dataset-content` inclui a `dataURI`, que é uma URL assinada para os resultados de saída. Tem validade por um curto período de tempo (algumas horas). Visite a URL `dataURI` para acessar os resultados da sua consulta SQL.

> **Note**
>
> Dependendo do seu fluxo de trabalho, você sempre pode chamar `get-dataset-content` antes de acessar o conteúdo, porque chamar esse comando gera uma nova URL assinada.

Tutorial: consultar AWS IoT SiteWise dados em AWS IoT Analytics

Este tutorial demonstra como consultar AWS IoT SiteWise dados em AWS IoT Analytics. O tutorial usa dados de uma demonstração AWS IoT SiteWise que fornece um conjunto de amostras de dados para um parque eólico.

> **Important**
>
> Você será cobrado pelos recursos que a demonstração criar e consumir.

Tópicos

- Pré-requisitos
- Carregar e verificar dados
- Exploração de dados
- Executar consultas estatísticas
- Limpeza de seus recursos do tutorial

Pré-requisitos

Para este tutorial, você precisa dos seguintes recursos:

- Você deve ter uma AWS conta para começar a usar AWS IoT SiteWise AWS IoT Analytics e. Se você ainda não possuir uma conta, siga os procedimentos em Criar uma conta da AWS.
- Um computador de desenvolvimento que executa Windows, macOS, Linux ou Unix para acessar o AWS Management Console. Para obter mais informações, consulte Conceitos básicos sobre o AWS Management Console.

- AWS IoT SiteWise dados que definem AWS IoT SiteWise modelos e ativos e transmitem dados que representam dados de equipamentos de parques eólicos. Para criar seus dados, siga as etapas em Criação da AWS IoT SiteWise demonstração no Guia do AWS IoT SiteWise usuário.
- Seus dados de AWS IoT SiteWise demonstração do equipamento do parque eólico em um armazenamento de dados existente que você gerencia. Para obter mais informações sobre como criar um armazenamento de dados para seus AWS IoT SiteWise dados, consulte Definir configurações de armazenamento no Guia AWS IoT SiteWise do usuário.

  > **Note**
  >
  > Seus AWS IoT SiteWise metadados aparecem em seu armazenamento de AWS IoT SiteWise dados logo após a criação; no entanto, pode levar até seis horas para que seus dados brutos apareçam. Enquanto isso, você pode criar um AWS IoT Analytics conjunto de dados e executar consultas nos seus metadados.

Próxima etapa

Carregar e verificar dados

Carregar e verificar dados

Os dados que você consulta neste tutorial são um conjunto de AWS IoT SiteWise dados de amostra que modela turbinas de motores eólicos em um parque eólico.

> **Note**
>
> Você consultará três tabelas em seu datastore ao longo deste tutorial:
>
> - `raw`: contém dados brutos e não processados para cada ativo.
> - `asset_metadata`: contém informações gerais sobre cada ativo.
> - `asset_hierarchy_metadata`: contém informações sobre as relações entre ativos.

Executar as consultas SQL neste tutorial

1. Siga as etapas em Crie um conjunto de dados com dados AWS IoT SiteWise (console) ou Criar um conjunto de dados com dados AWS IoT SiteWise (AWS CLI) para criar um AWS IoT Analytics conjunto de dados para seus AWS IoT SiteWise dados.

2. Para atualizar sua consulta de conjunto de dados ao longo deste tutorial, faça o seguinte.

   a. No AWS IoT Analytics console, na página Conjuntos de dados, escolha o nome do conjunto de dados que você criou na página anterior.

   b. Na página de resumo do conjunto de dados, escolha Editar para editar sua consulta SQL.

   c. Para exibir os resultados em uma tabela após a consulta, escolha Consulta de teste.

   Como alternativa, você pode executar o comando `update-dataset` a seguir para modificar a consulta SQL com a AWS CLI.

   ```
   aws iotanalytics update-dataset --cli-input-json file://update-query.json
   ```

   Conteúdo de `update-query.json`:

   ```
   {
       "datasetName": "my_dataset",
       "actions": [
           {
               "actionName": "myDatasetUpdateAction",
               "queryAction": {
                   "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
    LIMIT 3"
               }
           }
       ]
   }
   ```

3. No AWS IoT Analytics console ou com o AWS CLI, execute a consulta a seguir em seus dados para verificar se a `asset_metadata` tabela foi carregada com êxito.

   ```
   SELECT COUNT(*) FROM my_iotsitewise_datastore.asset_metadata
   ```

   Da mesma forma, você pode verificar se suas `asset_hierarchy_metadata` e tabelas `raw` não estão vazias.

Próxima etapa

Exploração de dados

Exploração de dados

Depois que seus AWS IoT SiteWise dados são criados e carregados em um armazenamento de dados, você pode criar um AWS IoT Analytics conjunto de dados e executar consultas SQL AWS IoT Analytics para descobrir insights sobre seus ativos. As consultas a seguir demonstram como você pode explorar seus dados antes de executar consultas estatísticas.

Para explorar seus dados com consultas SQL

1. Veja uma amostra de colunas e valores em cada tabela, como na tabela bruta.

   ```
   SELECT * FROM my_iotsitewise_datastore.raw LIMIT 5
   ```

2. Use `SELECT DISTINCT` para consultar sua `asset_metadata` tabela e listar os nomes (exclusivos) de seus AWS IoT SiteWise ativos.

   ```
   SELECT DISTINCT assetname FROM my_iotsitewise_datastore.asset_metadata ORDER BY
    assetname
   ```

3. Para listar informações sobre propriedades de um AWS IoT SiteWise ativo específico, use a `WHERE` cláusula.

   ```
   SELECT assetpropertyname,
        assetpropertyunit,
        assetpropertydatatype
   FROM my_iotsitewise_datastore.asset_metadata
   WHERE assetname = 'Demo Turbine Asset 2'
   ```

4. Com AWS IoT Analytics, você pode unir dados de duas ou mais tabelas em seu armazenamento de dados, como no exemplo a seguir.

   ```
   SELECT * FROM my_iotsitewise_datastore.raw AS raw
   JOIN my_iotsitewise_datastore.asset_metadata AS asset_metadata
   ON raw.seriesId = asset_metadata.timeseriesId
   ```

   Para visualizar todas as relações entre seus ativos, use a funcionalidade `JOIN` na consulta a seguir.

   ```
   SELECT DISTINCT parent.assetName as "Parent name",
       child.assetName AS "Child name"
   FROM (
   ```

```
    SELECT sourceAssetId AS parent,
            targetAssetId AS child
    FROM my_iotsitewise_datastore.asset_hierarchy_metadata
    WHERE associationType = 'CHILD'
)
AS relations
JOIN my_iotsitewise_datastore.asset_metadata AS child
    ON relations.child = child.assetId
JOIN my_iotsitewise_datastore.asset_metadata AS parent
    ON relations.parent = parent.assetId
```

Próxima etapa

Executar consultas estatísticas

Executar consultas estatísticas

Agora que você explorou seus AWS IoT SiteWise dados, pode executar consultas estatísticas que fornecem informações valiosas sobre seu equipamento industrial. As consultas a seguir demonstram algumas das informações que você pode recuperar.

Para executar consultas estatísticas sobre dados de AWS IoT SiteWise demonstração do parque eólico

1. Execute o seguinte comando SQL para encontrar os valores mais recentes de todas as propriedades com valores numéricos para um ativo específico (ativo da turbina de demonstração 4).

```
SELECT assetName,
    assetPropertyName,
    assetPropertyUnit,
    max_by(value, timeInSeconds) AS Latest
FROM (
    SELECT *,
        CASE assetPropertyDataType
        WHEN 'DOUBLE' THEN
        cast(doubleValue AS varchar)
        WHEN 'INTEGER' THEN
        cast(integerValue AS varchar)
        WHEN 'STRING' THEN
        stringValue
        WHEN 'BOOLEAN' THEN
```

```
        cast(booleanValue AS varchar)
        ELSE NULL
        END AS value
    FROM my_iotsitewise_datastore.asset_metadata AS asset_metadata
    JOIN my_iotsitewise_datastore.raw AS raw
        ON raw.seriesId = asset_metadata.timeSeriesId
    WHERE startYear=2021
        AND startMonth=7
        AND startDay=8
        AND assetName='Demo Turbine Asset 4'
)
GROUP BY assetName, assetPropertyName, assetPropertyUnit
```

2. Junte as tabelas de metadados e sua tabela bruta para identificar as propriedades máximas de velocidade do vento para todos os ativos, além dos ativos principais.

```
SELECT child_assets_data_set.parentAssetId,
        child_assets_data_set.childAssetId,
        asset_metadata.assetPropertyId,
        asset_metadata.assetPropertyName,
        asset_metadata.timeSeriesId,
        raw_data_set.max_speed
FROM (
   SELECT sourceAssetId AS parentAssetId,
        targetAssetId AS childAssetId
    FROM my_iotsitewise_datastore.asset_hierarchy_metadata
    WHERE associationType = 'CHILD'
)
AS child_assets_data_set
JOIN mls_demo.asset_metadata AS asset_metadata
    ON asset_metadata.assetId = child_assets_data_set.childAssetId
JOIN (
    SELECT seriesId, MAX(doubleValue) AS max_speed
    FROM my_iotsitewise_datastore.raw
    GROUP BY seriesId
)
AS raw_data_set
ON raw_data_set.seriesId = asset_metadata.timeseriesid
WHERE assetPropertyName = 'Wind Speed'
ORDER BY max_speed DESC
```

3. Para encontrar o valor médio de uma propriedade específica (Velocidade do Vento) para um ativo (ativo da turbina de demonstração 2), execute o seguinte comando SQL. Você deve substituir `my_bucket_id` pela ID do seu bucket.

```
SELECT AVG(doubleValue) as "Average wind speed"
FROM my_iotsitewise_datastore.raw
WHERE seriesId =
    (SELECT timeseriesId
    FROM my_iotsitewise_datastore.asset_metadata as asset_metadata
    WHERE asset_metadata.assetname = 'Demo Turbine Asset 2'
            AND asset_metadata.assetpropertyname = 'Wind Speed')
```

Próxima etapa

Limpeza de seus recursos do tutorial

Limpeza de seus recursos do tutorial

Depois de concluir o tutorial, limpe os recursos para evitar a geração de cobranças relacionadas.

Para excluir sua AWS IoT SiteWise demonstração

A AWS IoT SiteWise demonstração é excluída após uma semana. Se tiver terminado de usar os recursos de demonstração, você pode excluir a demonstração antes. Use as etapas a seguir para excluir a demonstração manualmente.

1. Navegue até o console do AWS CloudFormation.
2. Escolha IoTSiteWiseDemoAssets na lista de Pilhas.
3. Escolha Excluir. Quando você exclui a pilha, todos os recursos criados para a demonstração são excluídos.
4. No diálogo de confirmação, escolha Excluir.

   A pilha leva cerca de 15 minutos para ser excluída. Se houver falha na exclusão, escolha Excluir no canto superior direito novamente. Se a demonstração não for excluída novamente, siga as etapas no AWS CloudFormation console para ignorar os recursos que não foram excluídos e tente novamente.

Para excluir seu datastore

- Para excluir seu datastore gerenciado, execute o comando `delete-datastore` da CLI, como no exemplo a seguir.

```
aws iotanalytics delete-datastore --datastore-name my_IotSiteWise_datastore
```

Para excluir seu AWS IoT Analytics conjunto de dados

- Para excluir o conjunto de dados, execute o comando `delete-dataset` da CLI, como no exemplo a seguir. Você não precisa excluir o conteúdo do conjunto de dados antes de executar esta operação.

```
aws iotanalytics delete-dataset --dataset-name my_dataset
```

> **Note**
>
> Este comando não produz saída.

Atividades do pipeline

O pipeline funcional mais simples conecta um canal a um datastore, o que faz dele um pipeline com duas atividades: uma atividade `channel` e uma atividade `datastore`. Você pode alcançar um processamento de mensagens mais eficiente adicionando outras atividades ao pipeline.

Você pode usar a operação RunPipelineActivity para simular os resultados da execução de uma atividade de pipeline em uma carga útil de mensagem fornecida por você. Isso pode ser útil ao desenvolver e depurar atividades do seu pipeline. O exemplo de RunPipelineActivity demonstra como ele é usado.

Atividade Canal

A primeira atividade em um pipeline deve ser a atividade `channel` que determina a fonte das mensagens a serem processadas.

```
{
    "channel": {
        "name": "MyChannelActivity",
        "channelName": "mychannel",
        "next": "MyLambdaActivity"
    }
}
```

Atividade Datastore

A atividade `datastore`, que especifica onde armazenar os dados processados, é a última atividade.

```
{
    "datastore": {
        "name": "MyDatastoreActivity",
        "datastoreName": "mydatastore"
    }
}
```

Atividade AWS Lambda

Você pode usar uma atividade **`lambda`** para realizar processamento mais complexo na mensagem. Por exemplo, você pode enriquecer mensagens com dados da saída de operações externas de API ou filtrar mensagens com base na lógica do Amazon DynamoDB. No entanto, você não pode usar essa atividade de pipeline para adicionar mensagens adicionais ou remover mensagens existentes antes de entrar em um datastore.

A função AWS Lambda usada em uma atividade **`lambda`** deve receber e retornar uma matriz de objetos JSON. Para ver um exemplo, consulte the section called “Exemplo 1 da função do Lambda”.

Para conceder permissão AWS IoT Analytics para invocar a função do Lambda, você deve adicionar uma política. Por exemplo, execute o seguinte comando da CLI e substitua *`exampleFunctionName`* pelo nome da sua função do Lambda, substitua *`123456789012`* peloaID da conta da AWS e use o nome do recurso da Amazon (ARN) do pipeline que invoca a função do Lambda em questão.

```
aws lambda add-permission --function-name exampleFunctionName --
action lambda:InvokeFunction --statement-id iotanalytics --principal
 iotanalytics.amazonaws.com --source-account 123456789012 --source-arn
 arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline
```

O comando retorna o seguinte:

```
{
    "Statement": "{\"Sid\":\"iotanalyticsa\",\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"iotanalytics.amazonaws.com\"},\"Action\":
\"lambda:InvokeFunction\",\"Resource\":\"arn:aws:lambda:aws-region:aws-
account:function:exampleFunctionName\",\"Condition\":{\"StringEquals\":
{\"AWS:SourceAccount\":\"123456789012\"},\"ArnLike\":{\"AWS:SourceArn\":
\"arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline\"}}}"
}
```

Para mais informações, consulte Uso de políticas com base em recursos AWS Lambda em Guia do desenvolvedor AWS Lambda.

Exemplo 1 da função do Lambda

Neste exemplo, a função do Lambda adiciona informações com base nos dados da mensagem original. Um dispositivo publica uma mensagem com uma carga semelhante ao exemplo a seguir.

```
{
  "thingid": "00001234abcd",
  "temperature": 26,
  "humidity": 29,
  "location": {
    "lat": 52.4332935,
    "lon": 13.231694
  },
  "ip": "192.168.178.54",
  "datetime": "2018-02-15T07:06:01"
}
```

E o dispositivo tem a seguinte definição de pipeline.

```
{
    "pipeline": {
        "activities": [
            {
                "channel": {
                    "channelName": "foobar_channel",
                    "name": "foobar_channel_activity",
                    "next": "lambda_foobar_activity"
                }
            },
            {
                "lambda": {
                    "lambdaName": "MyAnalyticsLambdaFunction",
                    "batchSize": 5,
                    "name": "lambda_foobar_activity",
                    "next": "foobar_store_activity"
                }
            },
            {
                "datastore": {
                    "datastoreName": "foobar_datastore",
                    "name": "foobar_store_activity"
                }
            }
        ],
        "name": "foobar_pipeline",
        "arn": "arn:aws:iotanalytics:eu-west-1:123456789012:pipeline/foobar_pipeline"
    }
```

```
}
```

A seguinte função Lambda Python (`MyAnalyticsLambdaFunction`) adiciona a URL do GMaps e a temperatura em Fahrenheit à mensagem:

```
import logging
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def c_to_f(c):
    return 9.0/5.0 * c + 32

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))
    maps_url = 'N/A'

    for e in event:
        #e['foo'] = 'addedByLambda'
        if 'location' in e:
            lat = e['location']['lat']
            lon = e['location']['lon']
            maps_url = "http://maps.google.com/maps?q={},{}".format(lat,lon)

        if 'temperature' in e:
            e['temperature_f'] = c_to_f(e['temperature'])

        logger.info("maps_url: {}".format(maps_url))
        e['maps_url'] = maps_url

    logger.info("event after processing: {}".format(event))

    return event
```

Exemplo 2 da função do Lambda

Uma técnica útil é compactar e serializar cargas de mensagens para reduzir os custos de transporte e armazenamento. Neste segundo exemplo, a função do Lambda supõe que a carga da mensagem representa um JSON original que foi compactado e codificado em base64 (serializado) como uma string. Ela retorna o JSON original:

```
import base64
import gzip
import json
import logging
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def decode_to_bytes(e):
    return base64.b64decode(e)

def decompress_to_string(binary_data):
    return gzip.decompress(binary_data).decode('utf-8')

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))

    decompressed_data = []

    for e in event:
        binary_data = decode_to_bytes(e)
        decompressed_string = decompress_to_string(binary_data)

        decompressed_data.append(json.loads(decompressed_string))

    logger.info("event after processing: {}".format(decompressed_data))

    return decompressed_data
```

Atividade AddAttributes

Uma atividade `addAttributes` acrescenta atributos com base em atributos existentes na mensagem. Isso permite alterar a forma da mensagem antes que seja armazenada. Por exemplo, é possível usar `addAttributes` para normalizar dados vindos de diferentes gerações de firmware do dispositivo.

Considere a mensagem de entrada a seguir.

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6152543, -122.3354883 ]
    }
}
```

A atividade `addAttributes` é semelhante ao seguinte:

```
{
    "addAttributes": {
        "name": "MyAddAttributesActivity",
        "attributes": {
            "device.id": "id",
            "device.coord[0]": "lat",
            "device.coord[1]": "lon"
        },
        "next": "MyRemoveAttributesActivity"
    }
}
```

Essa atividade move a ID do dispositivo para o nível raiz e extrai os valores na matriz do `coord`, promovendo-os a atributos de nível superior chamados `lat` e `lon`. Como resultado dessa atividade, a mensagem de saída é convertida para o seguinte exemplo:

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6, -122.3 ]
    },
    "id": "device-123",
    "lat": 47.6,
```

```
    "lon": -122.3
}
```

O atributo de dispositivo original ainda está presente. Se quiser removê-lo, você pode usar a atividade `removeAttributes`.

Atividade RemoveAttributes

Uma atividade `removeAttributes` remove os atributos de uma mensagem. Por exemplo, considere a mensagem que foi o resultado da atividade `addAttributes`.

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6, -122.3 ]
    },
    "id": "device-123",
    "lat": 47.6,
    "lon": -122.3
}
```

Para normalizar essa mensagem de modo que ela inclua apenas os dados necessários no nível raiz, use a seguinte atividade `removeAttributes`:

```
{
    "removeAttributes": {
        "name": "MyRemoveAttributesActivity",
        "attributes": [
            "device"
        ],
        "next": "MyDatastoreActivity"
    }
}
```

Isso resulta na seguinte mensagem fluindo ao longo da pipeline:

```
{
    "id": "device-123",
    "lat": 47.6,
    "lon": -122.3
}
```

Atividade SelectAttributes

A atividade `selectAttributes` cria uma nova mensagem usando apenas os atributos especificados na mensagem original. Todos os outros atributos são descartados. `selectAttributes` cria novos atributos apenas na raiz da mensagem. Portanto, considere esta mensagem:

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6152543, -122.3354883 ],
        "temp": 50,
        "hum": 40
    },
    "light": 90
}
```

e esta atividade:

```
{
    "selectAttributes": {
        "name": "MySelectAttributesActivity",
        "attributes": [
            "device.temp",
            "device.hum",
            "light"
        ],
        "next": "MyDatastoreActivity"
    }
}
```

O resultado é a seguinte mensagem fluindo por meio do pipeline.

```
{
    "temp": 50,
    "hum": 40,
    "light": 90
}
```

Novamente, o `selectAttributes` só pode criar objetos no nível raiz.

Atividade Filtro

Uma atividade `filter` filtra uma mensagem com base em seus atributos. A expressão usada nessa atividade é semelhante a uma cláusula SQL `WHERE` que deve retornar um booleano.

```
{
    "filter": {
        "name": "MyFilterActivity",
        "filter": "temp > 40 AND hum < 20",
        "next": "MyDatastoreActivity"
    }
}
```

Atividade DeviceRegistryEnrich

A atividade `deviceRegistryEnrich` permite adicionar dados do registro de dispositivos do AWS IoT à carga útil da sua mensagem. Por exemplo, com base na seguinte mensagem:

```
{
    "temp": 50,
    "hum": 40,
    "device" {
        "thingName": "my-thing"
    }
}
```

e uma atividade `deviceRegistryEnrich` que será semelhante a esta:

```
{
    "deviceRegistryEnrich": {
        "name": "MyDeviceRegistryEnrichActivity",
        "attribute": "metadata",
        "thingName": "device.thingName",
        "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
        "next": "MyDatastoreActivity"
    }
}
```

A mensagem de saída é semelhante a este exemplo.

```
{
```

```
    "temp" : 50,
    "hum" : 40,
    "device" {
        "thingName" : "my-thing"
    },
    "metadata" : {
        "defaultClientId": "my-thing",
        "thingTypeName": "my-thing",
        "thingArn": "arn:aws:iot:us-east-1:<your-account-number>:thing/my-thing",
        "version": 1,
        "thingName": "my-thing",
        "attributes": {},
        "thingId": "aaabbbccc-dddeeef-gghh-jjkk-llmmnnoopp"
    }
}
```

Você deve especificar uma função no campo `roleArn` da definição da atividade que tenha as permissões apropriadas anexadas. A função deve ter uma política de permissões semelhante ao seguinte exemplo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeThing"
            ],
            "Resource": [
                "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
            ]
        }
    ]
}
```

e uma política de confiança semelhante a:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

Atividade DeviceShadowEnrich

Uma atividade `deviceShadowEnrich` adiciona informações do serviço de sombra do dispositivo de AWS IoT a uma mensagem. Por exemplo, considere a mensagem:

```
{
    "temp": 50,
    "hum": 40,
    "device": { "thingName": "my-thing" }
}
```

e a seguinte atividade `deviceShadowEnrich`:

```
{
    "deviceShadowEnrich": {
        "name": "MyDeviceShadowEnrichActivity",
        "attribute": "shadow",
        "thingName": "device.thingName",
        "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
        "next": "MyDatastoreActivity"
    }
}
```

O resultado é uma mensagem que parece com o exemplo a seguir.

```
{
    "temp": 50,
    "hum": 40,
    "device": {
        "thingName": "my-thing"
    },
```

```
    "shadow": {
        "state": {
            "desired": {
                "attributeX": valueX, ...
            },
            "reported": {
                "attributeX": valueX, ...
            },
            "delta": {
                "attributeX": valueX, ...
            }
        },
        "metadata": {
            "desired": {
                "attribute1": {
                    "timestamp": timestamp
                }, ...
            },
            "reported": ": {
                "attribute1": {
                    "timestamp": timestamp
                }, ...
            }
        },
        "timestamp": timestamp,
        "clientToken": "token",
        "version": version
    }
}
```

Você deve especificar uma função no campo `roleArn` da definição da atividade que tenha as permissões apropriadas anexadas. A função deve ter uma política de permissões semelhante à seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:GetThingShadow"
            ],
            "Resource": [
```

```
                "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
            ]
        }
    ]
}
```

e uma política de confiança semelhante a:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

Atividade Matemática

Uma atividade `math` calcula uma expressão aritmética usando os atributos da mensagem. A expressão deve retornar uma número. Por exemplo, considere a mensagem de entrada a seguir:

```
{
    "tempF": 50,
}
```

após o processamento pela seguinte atividade `math`:

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "(tempF - 32) / 2",
        "attribute": "tempC",
        "next": "MyDatastoreActivity"
```

```
    }
}
```

a mensagem resultante é semelhante a esta:

```
{
    "tempF" : 50,
    "tempC": 9
}
```

Funções e operadores de atividades matemáticas

É possível usar os seguintes operadores em uma atividade `math`:

+	adição
-	subtração
*	multiplicação
/	divisão
%	módulo

É possível usar as seguintes funções em uma atividade `math`:

- abs(Decimal)
- acos(Decimal)
- asin(Decimal)
- atan(Decimal)
- atan2(Decimal, Decimal)
- ceil(Decimal)
- cos(Decimal)
- cosh(Decimal)
- exp(Decimal)

- ln(Decimal)
- log(Decimal)
- mod(Decimal, Decimal)
- power(Decimal, Decimal)
- round(Decimal)
- sign(Decimal)
- sin(Decimal)
- sinh(Decimal)
- sqrt(Decimal)
- tan(Decimal)
- tanh(Decimal)
- trunc(Decimal, Integer)

abs(Decimal)

Gera o valor absoluto de um número.

Exemplos: `abs(-5)` retorna 5.

Tipo de argumento	Result
`Int`	`Int`, o valor absoluto do argumento.
`Decimal`	`Decimal`, o valor absoluto do argumento.
`Boolean`	`Undefined` .
`String`	`Decimal`. O resultado é o valor absoluto do argumento. Se a string não puder ser convertid a, o resultado será `Undefined` .
Array	`Undefined` .
Objeto	`Undefined` .
Nulo	`Undefined` .

Tipo de argumento	Result
Não definido	`Undefined` .

acos(Decimal)

Gera o cosseno inverso de um número em radianos. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `acos(0)` = 1,5707963267948966

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), o cosseno inverso do argumento. Os resultados imaginários são gerados como `Undefined` .
`Decimal`	`Decimal` (com precisão dupla), o cosseno inverso do argumento. Os resultados imaginários são gerados como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (com precisão dupla), o cosseno inverso do argumento. Se a string não puder ser convertida, o resultado será `Undefined` . Os resultados imaginários são gerados como `Undefined` .
Array	`Undefined` .
Objeto	`Undefined` .
Nulo	`Undefined` .
Não definido	`Undefined` .

asin(Decimal)

Gera o seno inverso de um número em radianos. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `asin(0)` = 0,0

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), o seno inverso do argumento. Os resultados imaginários são gerados como `Undefined` .
`Decimal`	`Decimal` (com precisão dupla), o seno inverso do argumento. Os resultados imaginários são gerados como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (com precisão dupla), o seno inverso do argumento. Se a string não puder ser convertida, o resultado será `Undefined` . Os resultados imaginários são gerados como `Undefined` .
Array	`Undefined` .
Objeto	`Undefined` .
Nulo	`Undefined` .
Não definido	`Undefined` .

atan(Decimal)

Gera a tangente inversa de um número em radianos. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `atan(0)` = 0,0

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), a tangente inversa do argumento. Os resultados imaginários são gerados como `Undefined` .
`Decimal`	`Decimal` (com precisão dupla), a tangente inversa do argumento. Os resultados imaginários são gerados como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (com precisão dupla), a tangente inversa do argumento. Se a string não puder ser convertida, o resultado será `Undefined` . Os resultados imaginários são gerados como `Undefined` .
Array	`Undefined` .
Objeto	`Undefined` .
Nulo	`Undefined` .
Não definido	`Undefined` .

atan2(Decimal, Decimal)

Gera o ângulo, em radianos, entre o eixo X positivo e o ponto (x, y) definido nos dois argumentos. O ângulo é positivo para os ângulos em sentido anti-horário (metade superior, y > 0) e negativo para os ângulos em sentido horário. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `atan(1, 0)` = 1,5707963267948966

Tipo de argumento	Tipo de argumento	Result
`Int` / `Decimal`	`Int` / `Decimal`	`Decimal` (com precisão dupla), o ângulo entre o eixo x e o ponto (x, y) especificado
`Int` / `Decimal` / `String`	`Int` / `Decimal` / `String`	`Decimal`, a tangente inversa do ponto descrito. Se uma string não puder ser convertida, o resultado será `Undefined` .
Outros valores	Outros valores	`Undefined` .

ceil(Decimal)

Arredonda o `Decimal` fornecido para o `Int` mais próximo.

Exemplos:

`ceil(1.2)` = 2

`ceil(11.2)` = -1

Tipo de argumento	Result
`Int`	`Int`, o valor do argumento.
`Decimal`	`Int`, a string será convertida em `Decimal` e arredondada para o mais próximo `Int`. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
Outros valores	`Undefined` .

cos(Decimal)

Gera o cosseno de um número em radianos. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `cos(0)` = 1

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), o cosseno do argumento. Os resultados imaginários são gerados como `Undefined` .
`Decimal`	`Decimal` (com precisão dupla), o cosseno do argumento. Os resultados imaginários são gerados como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (com precisão dupla), o cosseno do argumento. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` . Os resultados imaginários são gerados como `Undefined` .
Array	`Undefined` .
Objeto	`Undefined` .
Nulo	`Undefined` .
Não definido	`Undefined` .

cosh(Decimal)

Gera o cosseno hiperbólico de um número em radianos. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `cosh(2.3)` = 5,037220649268761

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), o cosseno hiperbólico do argumento. Os resultados imaginários são gerados como `Undefined` .
`Decimal`	`Decimal` (com precisão dupla), o cosseno hiperbólico do argumento. Os resultados imaginários são gerados como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (com precisão dupla), o cosseno hiperbólico do argumento. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` . Os resultados imaginários são gerados como `Undefined` .
Array	`Undefined` .
Objeto	`Undefined` .
Nulo	`Undefined` .
Não definido	`Undefined` .

exp(Decimal)

Retorna e elevado ao argumento decimal. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `exp(1)` = 1

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), argumento e ^.
`Decimal`	`Decimal` (com precisão dupla), argumento e ^.

Tipo de argumento	Result
`String`	`Decimal` (com precisão dupla), argumento e ^. Se a `String` não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
Outros valores	`Undefined` .

ln(Decimal)

Gera o logaritmo natural do argumento. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `ln(e)` = 1

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), o log natural do argumento.
`Decimal`	`Decimal` (com precisão dupla), o log natural do argumento.
`Boolean`	`Undefined` .
`String`	`Decimal` (com precisão dupla), o log natural do argumento. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
Array	`Undefined` .
Objeto	`Undefined` .
Nulo	`Undefined` .
Não definido	`Undefined` .

log(Decimal)

Gera o logaritmo na base 10 do argumento. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `log(100)` = 2,0

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), o log de base 10 do argumento.
`Decimal`	`Decimal` (com precisão dupla), o log de base 10 do argumento.
`Boolean`	`Undefined` .
`String`	`Decimal` (com precisão dupla), o log de base 10 do argumento. Se a `String` não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
Array	`Undefined` .
Objeto	`Undefined` .
Nulo	`Undefined` .
Não definido	`Undefined` .

mod(Decimal, Decimal)

Gera o restante da divisão do primeiro argumento pelo segundo argumento. Você também pode usar `%` como um operador infixo para a mesma funcionalidade do módulo.

Exemplos: `mod(8, 3)` = 3

Operando esquerdo	Operando direito	Resultado
`Int`	`Int`	`Int`, o primeiro argumento módulo do segundo argumento.
`Int` / `Decimal`	`Int` / `Decimal`	`Decimal`, o primeiro argumento módulo do segundo argumento.
`String` / `Int` / `Decimal`	`String` / `Int` / `Decimal`	Se todas as strings forem convertidas em `Decimals`, o resultado será o primeiro argumento como módulo do segundo argumento. Caso contrário, `Undefined` .
Outros valores	Outros valores	`Undefined` .

power(Decimal, Decimal)

Gera o primeiro argumento elevado ao segundo argumento. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `power(2, 5)` = 32,0

Tipo de argumento 1	Tipo de argumento 2	Resultado
`Int` / `Decimal`	`Int` / `Decimal`	Um `Decimal` (com precisão dupla), o primeiro argumento elevado para o poder do segundo argumento.
`Int` / `Decimal` / `String`	`Int` / `Decimal` / `String`	Um `Decimal` (com precisão dupla), o primeiro argumento elevado para o poder do segundo argumento

Tipo de argumento 1	Tipo de argumento 2	Resultado
		. Quaisquer strings são convertidas em `Decimals`. Se a conversão de alguma `String` em `Decimal` falhar, o resultado será `Undefined` .
Outros valores	Outros valores	`Undefined` .

round(Decimal)

Arredonda o `Decimal` fornecido para o `Int` mais próximo. Se o `Decimal` for equidistante de dois valores `Int` (por exemplo, 0,5), o `Decimal` será arredondado.

Exemplos:

`Round(1.2)` = 1

`Round(1.5)` = 2

`Round(1.7)` = 2

`Round(-1.1)` = -1

`Round(-1.5)` = -2

Tipo de argumento	Result
`Int`	O argumento
`Decimal`	`Decimal` é arredondado para o `Int` mais próximo.
`String`	`Decimal` é arredondado para o `Int` mais próximo. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
Outros valores	`Undefined` .

sign(Decimal)

Gera o sinal do número fornecido. Quando o sinal do argumento for positivo, 1 será gerado. Quando o sinal do argumento for negativo, -1 será gerado. Se o argumento for 0, 0 será gerado.

Exemplos:

`sign(-7)` = -1

`sign(0)` = 0

`sign(13)` = 1

Tipo de argumento	Result
`Int`	`Int`, o sinal do valor `Int`.
`Decimal`	`Int`, o sinal do valor `Decimal`.
`String`	`Int`, o sinal do valor `Decimal`. A string é convertida em um valor `Decimal`, e o sinal do valor `Decimal` é gerado. Se a `String` não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
Outros valores	`Undefined` .

sin(Decimal)

Gera o seno de um número em radianos. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `sin(0)` = 0,0

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), o seno do argumento.

Tipo de argumento	Result
`Decimal`	`Decimal` (com precisão dupla), o seno do argumento.
`Boolean`	`Undefined` .
`String`	`Decimal`, o seno do argumento. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

sinh(Decimal)

Gera o seno hiperbólico de um número em radianos. Valores `Decimal` são arredondados para dobrar a precisão antes da aplicação da função. O resultado é um valor `Decimal` de precisão dupla.

Exemplos: `sinh(2.3)` = 4,936961805545957

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), o seno hiperbólico do argumento.
`Decimal`	`Decimal` (com precisão dupla), o seno hiperbólico do argumento.
`Boolean`	`Undefined` .
`String`	`Decimal`, o seno hiperbólico do argumento. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` .

Tipo de argumento	Result
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

sqrt(Decimal)

Gera a raiz quadrada de um número. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `sqrt(9)` = 3,0

Tipo de argumento	Result
`Int`	A raiz quadrada do argumento.
`Decimal`	A raiz quadrada do argumento.
`Boolean`	`Undefined` .
`String`	A raiz quadrada do argumento. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

tan(Decimal)

Gera a tangente de um número em radianos. Valores `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplo: `tan(3)` = -0,1425465430742778

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), a tangente do argumento.
`Decimal`	`Decimal` (com precisão dupla), a tangente do argumento.
`Boolean`	`Undefined` .
`String`	`Decimal` (com precisão dupla), a tangente do argumento. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

tanh(Decimal)

Gera a tangente hiperbólica de um número em radianos. Valores `Decimal` são arredondados para dobrar a precisão antes da aplicação da função.

Exemplos: `tanh(2.3)` = 0,9800963962661914

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), a tangente hiperbólica do argumento.
`Decimal`	`Decimal` (com precisão dupla), a tangente hiperbólica do argumento.
`Boolean`	`Undefined` .
`String`	`Decimal` (com precisão dupla), a tangente hiperbólica do argumento. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

trunc(Decimal, Integer)

Trunca o primeiro argumento para o número de lugares `Decimal` especificado pelo segundo argumento. Se o segundo argumento for inferior a zero, ele será definida como zero. Se o segundo argumento for superior a 34, ele será definido como 34. Os zeros finais são removidos do resultado.

Exemplos:

`trunc(2.3, 0)` = 2

`trunc(2.3123, 2)` = 2,31

`trunc(2.888, 2)` = 2,88

`trunc(2.00, 5)` = 2

Tipo de argumento 1	Tipo de argumento 2	Result
`Int`	`Int`	O valor de origem.
`Int` / `Decimal` / `String`	`Int` / `Decimal`	O primeiro argumento é truncado para o comprimento descrito pelo segundo argumento. O segundo argumento, se não for um `Int`, será arredondado para o `Int` mais próximo. Strings são convertidas para valores `Decimal`. Se não for possível converter a string, o resultado será `Undefined` .
Outros valores		Indefinido.

RunPipelineActivity

Este é um exemplo de como você pode usar o comando `RunPipelineActivity` para testar uma atividade do pipeline. Para este exemplo, testamos uma atividade Math:

1. Crie um arquivo `maths.json` contendo a definição da atividade do pipeline que você deseja testar.

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "((temp - 32) * 5.0) / 9.0",
        "attribute": "tempC"
    }
}
```

2. Crie um arquivo `payloads.json` contendo as cargas de exemplo que são usadas para testar a atividade do pipeline.

```
[
```

```
    "{\"humidity\": 52, \"temp\": 68 }",
    "{\"humidity\": 52, \"temp\": 32 }"
]
```

3. Chame a operação `RunPipelineActivities` via linha de comando.

```
aws iotanalytics run-pipeline-activity  --pipeline-activity file://maths.json  --
payloads file://payloads.json --cli-binary-format raw-in-base64-out
```

Isso produz os seguintes resultados:

```
{
    "logResult": "",
    "payloads": [
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
    ]
}
```

As cargas listadas nos resultados são strings codificadas em Base64. Quando essas strings são decodificadas, você obtém os seguintes resultados:

```
{"humidity":52,"temp":68,"tempC":20}
{"humidity":52,"temp":32,"tempC":0}
```

Reprocessamento de mensagens do canal

AWS IoT Analytics permite que você reprocesse os dados do canal. Isso pode ser útil nos seguintes casos:

- Você quiser reproduzir dados consumidos em vez de iniciar novamente.
- Você fizer a atualização para um pipeline e quiser trazer dados existentes atualizados com as alterações.
- Você deseja incluir dados que foram ingeridos antes de fazer alterações nas opções de armazenamento gerenciado pelo cliente, nas permissões dos canais ou no armazenamento de dados.

Parâmetros

Ao reprocessar as mensagens do canal por meio do pipeline comAWS IoT Analytics, você deve especificar as seguintes informações:

`StartPipelineReprocessing`

Inicia o reprocessamento de mensagens por meio do pipeline.

`ChannelMessages`

Especifica um ou mais conjuntos de mensagens do canal que você deseja reprocessar.

Se você usar o objeto `channelMessages`, não deverá especificar um valor para `startTime` e `endTime`.

`s3Paths`

Especifica uma ou mais chaves que identificam os objetos do Amazon Simple Storage Service (Amazon S3) que salvam as mensagens do canal. Você deve usar o caminho completo para a chave.

Exemplo de caminho: `00:00:00/1582940490000_1582940520000_123456789012_mychannel_0_2118.0.jso`

Tipo: matriz de strings

Restrições de membros da matriz: 1 a 100 itens.

Restrições de comprimento: 1 a 1024 caracteres.

`endTime`

A hora de término (exclusivo) dos dados do canal que serão reprocessados.

Se você especificar um valor para o parâmetro `endTime`, não deverá usar o objeto `channelMessages`.

Tipo: Timestamp

`startTime`

A hora de início (inclusive) dos dados brutos da mensagem que serão reprocessados.

Se você especificar um valor para o parâmetro `startTime`, não deverá usar o objeto `channelMessages`.

Tipo: Timestamp

`pipelineName`

O nome do pipeline em que o reprocessamento será iniciado.

Tipo: sequência

Restrições de comprimento: 1 a 128 caracteres.

Reprocessar mensagens do canal (console)

Este tutorial mostra como reprocessar os dados do canal que estão armazenados no objeto Amazon S3 especificado no console AWS IoT Analytics.

Antes de começar, certifique-se de que as mensagens do canal que pretende reprocessar estão salvas em um bucket do Amazon S3 gerenciado pelo cliente.

1. Faça login no console do AWS IoT Analytics.
2. No painel de navegação, selecione Pipelines.
3. Selecione seu pipeline de destino.

4. Escolha Reprocessar mensagens em Ações.
5. Na página de reprocessamento do pipeline, escolha objetos do S3 para reprocessar mensagens.

   O console AWS IoT Analytics também oferece as seguintes opções:

   - Todo o intervalo disponível — reprocesse todos os dados válidos no canal.
   - Últimos 120 dias — reprocesse os dados que chegaram nos últimos 120 dias.
   - Últimos 90 dias — reprocesse os dados que chegaram nos últimos 90 dias.
   - Últimos 30 dias — reprocesse os dados que chegaram nos últimos 30 dias.
   - Intervalo personalizado — reprocesse os dados que chegaram no intervalo de tempo especificado. Você pode escolher qualquer intervalo de tempo.
6. Insira a chave do objeto Amazon S3 que armazena as mensagens do seu canal.

   Para encontrar a chave, faça o seguinte:

   a. Acesse o console do Amazon S3.
   b. Escolha o objeto do Amazon S3 de destino.
   c. Em Propriedades, na seção Visão geral do objeto, copie a chave.
7. Escolha Iniciar reprocessamento.

Reprocessamento de mensagens do canal (API)

Ao usar a API `StartPipelineReprocessing`, observe o seguinte:

- Os parâmetros `startTime` e `endTime` especificam quando os dados brutos foram consumidos, mas esses são cálculos genéricos. É possível arredondar para a hora mais próxima. O `startTime` é inclusivo, mas `endTime` é exclusivo.
- O comando inicia o reprocessamento de forma assíncrona e retorna imediatamente.
- Não há garantia de que as mensagens reprocessadas são processadas na ordem em que foram recebidas originalmente. Elas são aproximadamente as mesmas, mas não exatamente.
- Você pode fazer até 1.000 solicitações da API `StartPipelineReprocessing` a cada 24 horas para reprocessar as mensagens do mesmo canal por meio de um pipeline.
- O reprocessamento dos dados brutos incorre em custos adicionais.

Para obter mais informações, consulte a API StartPipelineReprocessing na AWS IoT AnalyticsReferência de API.

Cancelamento de atividades de reprocessamento de canais

Para cancelar uma atividade de reprocessamento de pipeline, use a API CancelPipelineReprocessing ou escolha Cancelar reprocessamento na página Atividades no console AWS IoT Analytics. Se você cancelar o reprocessamento, os dados restantes não serão reprocessados. Você deve iniciar outra solicitação de reprocessamento.

Use a API DescribePipeline para verificar o status do reprocessamento. Consulte o campo `reprocessingSummaries` na resposta.

Automação de seu fluxo de trabalho

AWS IoT Analytics fornece análise avançada de dados para AWS IoT. É possível coletar dados da IoT, processá-los, armazená-los e analisá-los automaticamente usando as ferramentas de aprendizagem profunda e de análise de dados. É possível executar contêineres que hospedam seu próprio código analítico personalizado ou caderno Jupyter ou usar contêineres de código personalizado de terceiros para que não seja necessário recriar ferramentas de análise existentes. É possível usar os seguintes recursos para coletar dados de entrada de um datastore e alimentá-los em um fluxo de trabalho automatizado:

Criar conteúdo do conjunto de dados em uma programação recorrente

Programe a criação automática do conteúdo do conjunto de dados especificando um gatilho ao chamar `CreateDataset` (`triggers:schedule:expression`). Os dados que estão em um datastore são usados para criar o conteúdo do conjunto de dados. É possível selecionar os campos que você deseja usando uma consulta SQL (`actions:queryAction:sqlQuery`).

Defina um período contíguo não sobreposto para garantir que o novo conjunto de dados contenha somente os dados recebidos desde a última vez. Use os campos `actions:queryAction:filters:deltaTime` e `:offsetSeconds` para especificar o período delta. Depois, especifique um gatilho para criar o conteúdo do conjunto de dados quando o intervalo de tempo tiver decorrido. Consulte the section called “Exemplo 6: criação de um conjunto de dados SQL com uma janela delta (CLI)”.

Criar conteúdo do conjunto de dados após a conclusão de outro conjunto de dados

Acione a criação de conteúdo do conjunto de dados quando outra criação de conteúdo do conjunto de dados for concluída `triggers:dataset:name`.

Executar a análise de seus aplicativos automaticamente

Conteinerize seus próprios aplicativos de análise de dados personalizados e acione-os para serem executados quando outro conteúdo do conjunto de dados for criado. Dessa maneira, é possível alimentar o aplicativo com dados do conteúdo do conjunto de dados que é criado em uma programação recorrente. É possível realizar uma ação automaticamente com relação aos resultados da análise no aplicativo. (`actions:containerAction`)

Criar conteúdo do conjunto de dados após a conclusão de outro conjunto de dados

Acione a criação de conteúdo do conjunto de dados quando outra criação de conteúdo do conjunto de dados for concluída `triggers:dataset:name`.

Executar a análise de seus aplicativos automaticamente

Conteinerize seus próprios aplicativos de análise de dados personalizados e acione-os para serem executados quando outro conteúdo do conjunto de dados for criado. Dessa maneira, é possível alimentar o aplicativo com dados do conteúdo do conjunto de dados que é criado em uma programação recorrente. É possível realizar uma ação automaticamente com relação aos resultados da análise no aplicativo. (`actions:containerAction`)

Casos de uso

Automatizar a medição da qualidade do produto para reduzir a OpEx

Você tem um sistema com uma válvula inteligente que mede a pressão, a umidade e a temperatura. O sistema coleta eventos periodicamente e quando determinados eventos ocorrem, como quando uma válvula abre e fecha. Com o AWS IoT Analytics, é possível automatizar uma análise que agrega dados não sobrepostos dessas janelas periódicas e cria relatórios de KPI sobre a qualidade do produto final. Depois de processar cada lote, você mede a qualidade geral do produto e reduz suas despesas operacionais por meio do volume de execução maximizado.

Automatizar a análise de uma frota de dispositivos

Você executa análises (algoritmo, ciência de dados ou ML para KPI) a cada 15 minutos em dados gerados por centenas de dispositivos. Com cada ciclo de análise gerando e armazenando o estado para a próxima execução da análise. Para cada uma de suas análises, você quer usar apenas os dados recebidos em um período especificado. Com o AWS IoT Analytics, você pode orquestrar suas análises e criar o KPI e o relatório de cada execução e, em seguida, armazenar os dados para análise futura.

Automatizar a detecção de anomalias

O AWS IoT Analytics permite automatizar seu fluxo de trabalho de detecção de anomalias que você precisa executar manualmente a cada 15 minutos em novos dados recebidos em um datastore. Você também pode automatizar um painel para mostrar o uso de dispositivos e os principais usuários em um período especificado.

Predizer resultados do processo industrial

Você tem linhas de produção industriais. Usando os dados enviados ao AWS IoT Analytics, incluindo medições dos processos disponíveis, você pode operacionalizar os fluxos de trabalho analíticos para prever os resultados do processo. Os dados do modelo podem ser organizados

em uma matriz M x N, onde cada linha contém dados de vários pontos de tempo em que as amostras de laboratório são coletadas. AWS IoT Analytics ajuda você a operacionalizar seu fluxo de trabalho analítico criando janelas delta e usando suas ferramentas de ciência de dados para criar KPIs e salvar o estado dos dispositivos de medição.

Como usar um contêiner do Docker

Esta seção inclui informações sobre como criar seu próprio contêiner do Docker. Há um risco de segurança caso você use novamente contêineres do Docker criados por terceiros: esses contêineres podem executar um código arbitrário com suas permissões de usuário. Verifique se você confia no autor de qualquer contêiner de terceiros antes de usá-lo.

Estas são as etapas para configurar a análise de dados periódica em dados recebidos desde a última análise executada:

1. Crie um contêiner de docker que contenha seu aplicativo de dados mais todas as bibliotecas necessárias ou outras dependências.

   A extensão do Jupyter do IotAnalytics fornece uma API de conteinerização para auxiliar no processo de conteinerização. Você também pode executar imagens de sua própria criação, nas quais cria ou monta o conjunto de ferramentas do aplicativo para realizar a análise ou o cálculo de dados desejados. AWS IoT Analytics permite que você defina a origem dos dados de entrada para o aplicativo em contêiner e o destino dos dados de saída do contêiner do Docker por meio de variáveis. (Variáveis de entrada/saída do contêiner do docker personalizado contêm mais informações sobre o uso de variáveis com um contêiner personalizado.)
2. Faça upload do contêiner em um registro do Amazon ECR.
3. Crie um datastore para receber e armazenar mensagens (dados) de dispositivos (`iotanalytics: CreateDatastore`)
4. Crie um canal para o qual as mensagens sejam enviadas (`iotanalytics: CreateChannel`).
5. Crie um pipeline para conectar o canal ao datastore (`iotanalytics: CreatePipeline`).
6. Crie um perfil do IAM que conceda permissão para enviar dados da mensagem a um canal AWS IoT Analytics (`iam: CreateRole`.)
7. Crie uma regra de IoT que use uma consulta SQL para conectar um canal à origem dos dados da mensagem (campo `iot: CreateTopicRule topicRulePayload:actions:iotAnalytics`). Quando um dispositivo envia uma mensagem com o tópico apropriado por MQTT, ela é roteada para o canal. Ou você pode usar

o `iotanalytics: BatchPutMessage` para enviar mensagens diretamente para um canal de um dispositivo que pode usar o SDK da AWS ou AWS CLI.

8. Crie um conjunto de dados SQL cuja criação seja acionada por uma programação (campo `iotanalytics: CreateDataset,` `actions: queryAction:sqlQuery`).

   Você também especifica um filtro a ser aplicado aos dados da mensagem para ajudar a limitar as mensagens àquelas que chegaram desde a última execução da ação. (O campo `actions:queryAction:filters:deltaTime:timeExpression` fornece uma expressão pela qual a hora de uma mensagem pode ser determinada, enquanto o campo `actions:queryAction:filters:deltaTime:offsetSeconds` especifica a latência possível na chegada de uma mensagem.)

   O pré-filtro, juntamente com a programação do acionador, determina a “janela delta”. Cada novo conjunto de dados SQL é criado usando as mensagens recebidas desde a última vez em que o conjunto de dados SQL foi criado. (E quanto à primeira vez em que o conjunto de dados SQL é criado? Uma estimativa de quando o conjunto de dados teria sido criado pela última vez é feita de acordo com a programação e o pré-filtro.)

9. Crie outro conjunto de dados que seja acionado pela criação do primeiro (campo `trigger:dataset` CreateDataset). Para esse conjunto de dados, especifique uma ação de contêiner (campo `actions:containerAction`) que aponte e forneça informações necessárias para executar, o contêiner docker que você criou na primeira etapa. Aqui você também especifica:

   - O ARN do contêiner do Docker armazenado em sua conta (`image`).
   - O ARN da função que dá permissão ao sistema para acessar os recursos necessários para executar a ação do contêiner (`executionRoleArn`).
   - A configuração do recurso que executa a ação do contêiner (`resourceConfiguration`).
   - O tipo do recurso computacional usado para executar a ação do contêiner (`computeType` com valores possíveis: `ACU_1 [vCPU=4, memory=16GiB] or ACU_2 [vCPU=8, memory=32GiB]`).
   - O tamanho (em GB) do armazenamento persistente disponível para a instância do recurso usado para executar a ação do contêiner (`volumeSizeInGB`).
   - Os valores das variáveis usadas no contexto da execução do aplicativo contido (basicamente, os parâmetros passados para o aplicativo) (`variables`).

Essas variáveis são substituídas no momento da execução de um contêiner. Isso permite que você execute o mesmo contêiner com diferentes variáveis (parâmetros) que são fornecidas no momento em que o conteúdo do conjunto de dados é criado. A extensão do Jupyter do IotAnalytics simplifica esse processo reconhecendo automaticamente as variáveis em um notebook e disponibilizando-as como parte do processo de conteinerização. Você pode escolher as variáveis reconhecidas ou adicionar suas próprias variáveis personalizadas. Antes de executar um contêiner, o sistema substitui cada uma dessas variáveis pelo valor atual no momento da execução.

- Uma das variáveis é o nome do conjunto de dados cujo conteúdo mais recente é usado como entrada para o aplicativo (esse é o nome do conjunto de dados que você criou na etapa anterior) (`datasetContentVersionValue:datasetName`).

Com a consulta SQL e a janela delta para gerar o conjunto de dados e o contêiner com seu aplicativo, o AWS IoT Analytics cria um conjunto de dados de produção programado que é executado no intervalo especificado nos dados da janela delta, produzindo a saída desejada e enviando notificações.

Você pode pausar o aplicativo do conjunto de dados de produção e retomá-lo sempre que optar por fazê-lo. Quando você retoma o aplicativo do conjunto de dados de produção, o AWS IoT Analytics, por padrão, coleta todos os dados que chegaram desde a última execução, mas que ainda não foram analisados. Você também pode configurar como deseja retomar seu conjunto de dados de produção (tamanho da janela de trabalho) executando uma série de execuções consecutivas. Como alternativa, você pode retomar o aplicativo do conjunto de dados de produção, capturando apenas os dados recém-chegados que se ajustam ao tamanho especificado de sua janela delta.

Observe as seguintes limitações ao criar ou definir um conjunto de dados que é acionado pela criação de outro conjunto de dados:

- Somente conjuntos de dados de contêiner podem ser acionados por conjuntos de dados SQL.
- Um conjunto de dados SQL pode acionar, no máximo, 10 conjuntos de dados de contêiner.

Os seguintes erros podem ser retornados ao criar um conjunto de dados de contêiner que é acionado por um conjunto de dados SQL:

- "O conjunto de dados de acionamento só pode ser adicionado em um conjunto de dados de contêiner"

- "Pode haver somente um conjunto de dados de acionamento"

  Esse erro ocorre quando você tenta definir um conjunto de dados de contêiner que é acionado por dois conjuntos de dados SQL diferentes.

- “O conjunto de dados de acionamento <dataset-name> não pode ser acionado por um conjunto de dados de contêiner”

  Esse erro ocorre quando você tenta definir um conjunto de dados de contêiner que é acionado por outro conjunto de dados de contêiner.

- “<N> conjuntos de dados já são dependentes do conjunto de dados <dataset-name>”.

  Esse erro ocorre ao tentar definir outro conjunto de dados de contêiner que é acionado por um conjunto de dados SQL que já aciona 10 conjuntos de dados de contêiner.

- "Exatamente um tipo de trigger deve ser fornecido"

  Esse erro ocorre quando você tenta definir um conjunto de dados que é acionado por um trigger de programação e por um trigger de conjunto de dados.

Variáveis de entrada/saída do contêiner docker personalizado

Esta seção demonstra como o programa que é executado por sua imagem de docker personalizada pode ler variáveis de entrada e fazer upload de sua saída.

Arquivo de parâmetros

As variáveis de entrada e os destinos nos quais você deseja fazer upload da saída são armazenados em um arquivo JSON localizado em `/opt/ml/input/data/iotanalytics/params` na instância que executa a imagem do Docker. Este é um exemplo do conteúdo desse arquivo.

```
{
   "Context": {
       "OutputUris": {
           "html": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.html",
           "ipynb": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.ipynb"
       }
   },
   "Variables": {
```

```
       "source_dataset_name": "mydataset",
       "source_dataset_version_id": "xxxx",
       "example_var": "hello world!",
       "custom_output": "s3://aws-iot-analytics/dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.txt"
   }
}
```

Além do nome e do ID da versão do seu conjunto de dados, a seção `Variables` contém as variáveis especificadas na invocação de `iotanalytics:CreateDataset` - neste exemplo, uma variável `example_var` recebeu o valor `hello world!`. Um URI de saída personalizado também foi fornecido na variável `custom_output`. O campo `OutputUris` contém os locais padrão nos quais o contêiner pode fazer upload de sua saída -- neste exemplo, os URIs de saída padrão foram fornecidos para a saída html e ipynb.

Variáveis de entrada

O programa iniciado por sua imagem de docker pode ler variáveis no arquivo `params`. Este é um programa de exemplo que abre o arquivo `params`, analisa-o e imprime o valor da variável `example_var`.

```
import json

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
example_var = params["Variables"]["example_var"]
print(example_var)
```

Upload da saída

O programa iniciado por sua imagem do Docker também pode armazenar sua saída em um local do Amazon S3. A saída deve ser carregada com uma lista de controle de acesso "bucket-owner-full-control". A lista de acesso concede ao serviço do AWS IoT Analytics controle sobre a saída por upload. Neste exemplo, estendemos a anterior para fazer upload do conteúdo de `example_var` no local do Amazon S3 definido por `custom_output` no arquivo `params`.

```
import boto3
import json
from urllib.parse import urlparse
```

```
ACCESS_CONTROL_LIST = "bucket-owner-full-control"

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
example_var = params["Variables"]["example_var"]

outputUri = params["Variables"]["custom_output"]
# break the S3 path into a bucket and key
bucket = urlparse(outputUri).netloc
key = urlparse(outputUri).path.lstrip("/")

s3_client = boto3.client("s3")
s3_client.put_object(Bucket=bucket, Key=key, Body=example_var, ACL=ACCESS_CONTROL_LIST)
```

Permissões

É necessário criar duas funções do . Uma função concede permissão para iniciar uma instância do SageMaker para conteinerizar um notebook. Outra função é necessária para executar um contêiner.

Você pode criar a primeira função de forma manual ou automática. Se criar sua nova instância do Amazon SageMaker com o console do AWS IoT Analytics, você terá a opção de criar automaticamente uma nova função que concede todos os privilégios necessários para executar instâncias do SageMaker e conteinerizar cadernos. Ou você pode criar uma função com esses privilégios manualmente. Para fazer isso, crie uma função com a política `AmazonSageMakerFullAccess` anexada e adicione a seguinte política:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchDeleteImage",
        "ecr:BatchGetImage",
        "ecr:CompleteLayerUpload",
        "ecr:CreateRepository",
        "ecr:DescribeRepositories",
        "ecr:GetAuthorizationToken",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
```

```
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::iotanalytics-notebook-containers/*"
    }
  ]
}
```

Você deve criar manualmente a segunda função que concede permissão para executar um contêiner. Você deve fazer isso mesmo que tenha usado o console do AWS IoT Analytics para criar a primeira função automaticamente. Crie uma função com a política a seguir e a política de confiança anexadas:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:PutObject",
                "s3:GetObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::aws-*-dataset-*/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iotanalytics:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:GetDownloadUrlForLayer",
```

```
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
    ]
}
```

Veja a seguir um exemplo de política de confiança.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": ["sagemaker.amazonaws.com", "iotanalytics.amazonaws.com"]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Usando a API CreateDataset via Java e o AWS CLI

Cria um conjunto de dados. Um conjunto de dados armazena dados recuperados de um datastore aplicando uma `queryAction` (uma consulta SQL) ou uma `containerAction` (executando

uma aplicação em contêiner). Esta operação cria o esqueleto de um conjunto de dados. O conjunto de dados pode ser preenchido manualmente chamando `CreateDatasetContent` ou automaticamente, de acordo com um `trigger` que você especificar. Para obter mais informações, consulte CreateDataset e `CreateDatasetContent`.

Tópicos

- Exemplo 1: criação de um conjunto de dados SQL (java)
- Exemplo 2: criação de um conjunto de dados SQL com uma janela delta (java)
- Exemplo 3: criação de um conjunto de dados de contêiner com seu próprio trigger de programação (java)
- Exemplo 4: criação de um conjunto de dados de contêiner com um conjunto de dados SQL como um trigger (java)
- Exemplo 5: criação de um conjunto de dados SQL (CLI)
- Exemplo 6: criação de um conjunto de dados SQL com uma janela delta (CLI)

Exemplo 1: criação de um conjunto de dados SQL (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("SQLAction1");
action.setQueryAction(new SqlQueryDatasetAction().withSqlQuery("select * from
 DataStoreName"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);
```

```
// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Saída com êxito:

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

Exemplo 2: criação de um conjunto de dados SQL com uma janela delta (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Filter for DeltaTime
QueryFilter deltaTimeFilter = new QueryFilter();
deltaTimeFilter.withDeltaTime(
                new DeltaTime()
                .withOffsetSeconds(-1 * EstimatedDataDelayInSeconds)
                .withTimeExpression("from_unixtime(timestamp)"));

//Create Action
action.setActionName("SQLActionWithDeltaTime");
action.setQueryAction(new SqlQueryDatasetAction()
                .withSqlQuery("SELECT * from DataStoreName")
                .withFilters(deltaTimeFilter));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
```

```
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Saída com êxito:

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

Exemplo 3: criação de um conjunto de dados de contêiner com seu próprio trigger de programação (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
        .withImage(ImageURI)
        .withExecutionRoleArn(ExecutionRoleArn)
        .withResourceConfiguration(
                new ResourceConfiguration()
                .withComputeType(new ComputeType().withAcu(1))
                .withVolumeSizeInGB(1))
        .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
```

```
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Saída com êxito:

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

Exemplo 4: criação de um conjunto de dados de contêiner com um conjunto de dados SQL como um trigger (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
        .withImage(ImageURI)
        .withExecutionRoleArn(ExecutionRoleArn)
        .withResourceConfiguration(
                new ResourceConfiguration()
                .withComputeType(new ComputeType().withAcu(1))
                .withVolumeSizeInGB(1))
        .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);
```

```
//Create Trigger
DatasetTrigger trigger = new DatasetTrigger()
        .withDataset(new TriggeringDataset()
                .withName(TriggeringSQLDataSetName));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);
final CreateDatasetResult result = iot.createDataset(request);
```

Saída com êxito:

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>}
```

Exemplo 5: criação de um conjunto de dados SQL (CLI)

```
aws iotanalytics --endpoint <EndPoint>  --region <Region> create-dataset --dataset-
name="<dataSetName>" --actions="[{\"actionName\":\"<ActionName>\", \"queryAction\":
{\"sqlQuery\":\"<SQLQuery>\"}}]" --retentionPeriod numberOfDays=10
```

Saída com êxito:

```
{
    "datasetName": "<datasetName>",
    "datasetArn": "<datatsetARN>",
    "retentionPeriod": {unlimited: true} or {numberOfDays: 10, unlimited: false}
}
```

Exemplo 6: criação de um conjunto de dados SQL com uma janela delta (CLI)

Janelas delta são uma série de períodos definidos pelo usuário, intervalos não sobrepostos e contínuos. As janelas delta permitem que você crie conteúdo de conjunto de dados e execute a análise de dados novos recebidos no datastore desde a última análise. Você cria um janela

delta configurando o `deltaTime` na parte `filters` de uma `queryAction` de um conjunto de dados (CreateDataset). Geralmente, o conteúdo do conjunto de dados é criado automaticamente ao configurar também um gatilho de intervalo de tempo (`triggers:schedule:expression`). Basicamente, isso permite que você filtre as mensagens que chegaram durante um período específico, para que os dados contidos nas mensagens dos períodos anteriores não sejam contados duas vezes.

Neste exemplo, criamos um conjunto de dados que cria automaticamente conteúdo do conjunto de dados a cada 15 minutos usando somente esses dados que chegaram desde a última vez. Especificamos um desvio `deltaTime` de três minutos (180 segundos) que permite um atraso de três minutos para que as mensagens cheguem no datastore especificado. Portanto, se o conteúdo do conjunto de dados é criado às 10h30, os dados usados (incluídos no conteúdo do conjunto de dados) seriam aqueles com timestamps entre 10h12 e 10h27 (ou seja, 10h30 – 15 minutos – 3 minutos até 10h30 – 3 minutos).

```
aws iotanalytics --endpoint <EndPoint>  --region <Region> create-dataset --cli-input-
json file://delta-window.json
```

Onde o arquivo `delta-window.json` contém o código a seguir.

```
{
  "datasetName": "delta_window_example",
  "actions": [
    {
      "actionName": "delta_window_action",
      "queryAction": {
        "sqlQuery": "SELECT temperature, humidity, timestamp FROM my_datastore",
        "filters": [
          {
            "deltaTime": {
              "offsetSeconds": -180,
              "timeExpression": "from_unixtime(timestamp)"
            }
          }
        ]
      }
    }
  ],
  "triggers": [
    {
      "schedule": {
```

```
        "expression": "cron(0/15 * * * ? *)"
      }
    }
  ]
}
```

Saída com êxito:

```
{
    "datasetName": "<datasetName>",
    "datasetArn": "<datatsetARN>",
}
```

Conteinerização de caderno

Esta seção inclui informações sobre como criar um contêiner do Docker usando um caderno Jupyter. Há um risco de segurança se você usar blocos de anotações criados por terceiros: os contêineres incluídos poderão executar código arbitrário com as permissões de usuário. Além disso, o HTML gerado pelo bloco de anotações pode ser exibido em um console do AWS IoT Analytics, fornecendo um possível vetor de ataque no computador que está exibindo o HTML. Certifique-se de confiar no autor de qualquer bloco de anotações de terceiros antes de usá-lo.

Uma opção para executar funções analíticas avançadas é usar um Notebook Jupyter. O caderno Jupyter fornece poderosas ferramentas de ciência de dados que podem realizar machine learning e uma ampla variedade de análises estatísticas. Para obter mais informações, consulte Modelos de caderno. (Observe que, no momento, não oferecemos suporte à conteinerização dentro do JupyterLab.) Você pode empacotar seus cadernos Jupyter e bibliotecas em um contêiner que executa periodicamente em um novo lote de dados à medida que são recebidos pelo AWS IoT Analytics durante um período delta definido por você. Você pode programar um trabalho de análise que usa o contêiner e os novos dados segmentados capturados na janela de tempo especificada e armazenar a saída do trabalho para futuras análises programadas.

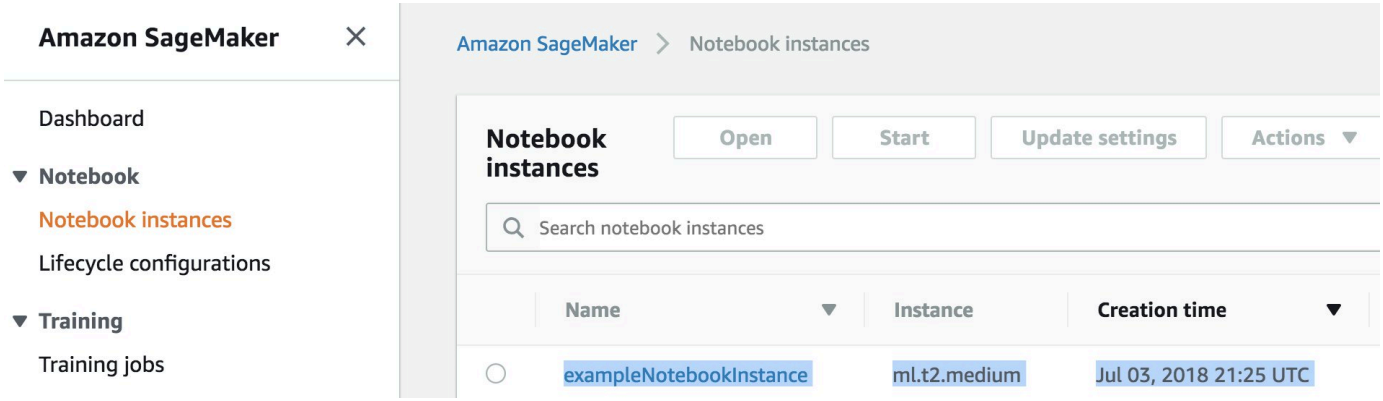
Se você tiver criado uma instância do SageMaker usando o console AWS IoT Analytics depois de 23 de agosto de 2018, a instalação da extensão da conteinerização terá sido feita para você automaticamente e é possível começar a criar uma imagem em contêineres. Caso contrário, siga as etapas listadas nesta seção para habilitar a conteinerização do notebook em sua instância do SageMaker. Na sequência, modifique a função de execução do SageMaker, para permitir que você faça upload da imagem do contêiner no Amazon EC2 e instale extensão da conteinerização.

Habilitar a conteinerização de instâncias de cadernos não criadas pelo console AWS IoT Analytics

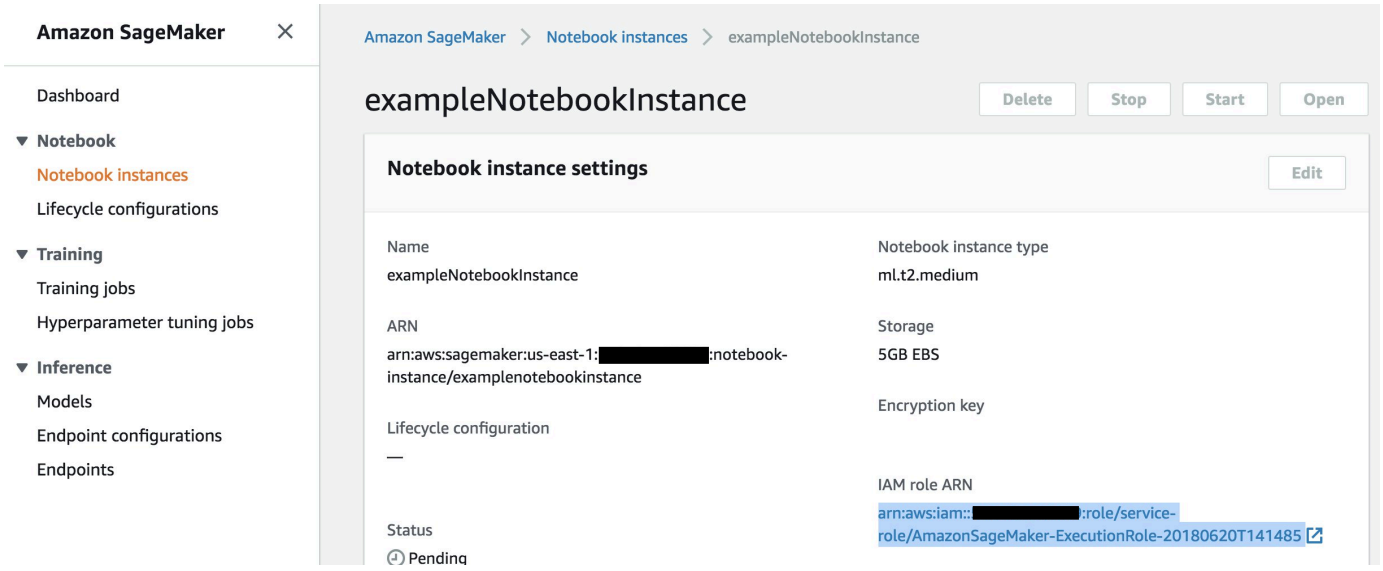
Recomendamos criar uma nova instância do SageMaker por meio do console AWS IoT Analytics em vez de seguir estas etapas. As novas instâncias oferecem suporte à conteinerização automaticamente.

Se você reiniciar a instância do SageMaker após habilitar a conteinerização conforme mostrado aqui, não será necessário adicionar novamente as políticas e funções do IAM, mas você deverá reinstalar a extensão, conforme mostrado na etapa final.

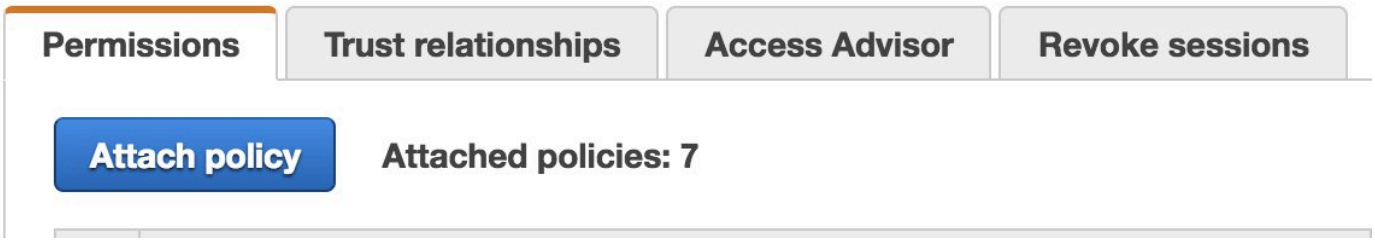
1. Para conceder acesso a sua instância de caderno ao Amazon ECS, selecione a instância do SageMaker na página do SageMaker:



2. Em ARN do perfil do IAM escolha a Função de execução do SageMaker.

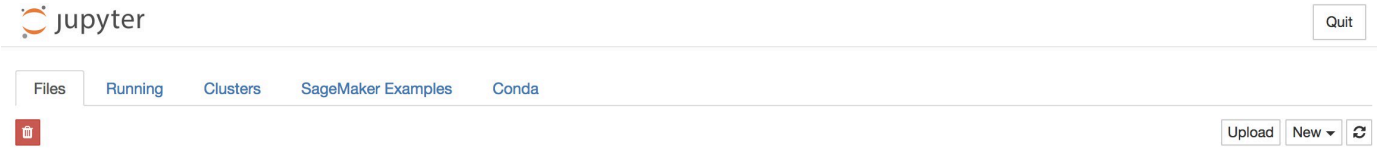


3. Escolha Attach Policy (Anexar política) e, em seguida, defina e anexe a política mostrada em Permissões. Se a política `AmazonSageMakerFullAccess` ainda não foi anexada, anexe-a.



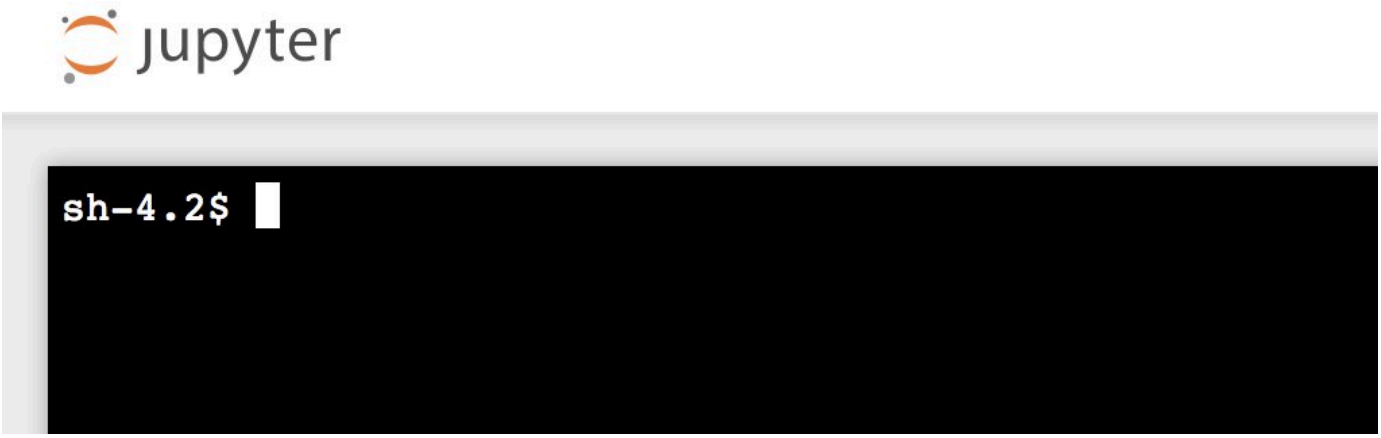
Você também deve baixar o código de conteinerização do Amazon S3 e instalá-lo na instância do seu caderno. A primeira etapa é acessar o terminal da instância do SageMaker.

1. Dentro do Jupyter, escolha Novo:



2. No menu exibido, escolha Terminal.

3. No terminal, digite os seguintes comandos para fazer download do código, descompactá-lo e instalá-lo. Observe que esses comandos eliminam todos os processos que estão sendo executados pelos blocos de anotações nessa instância do SageMaker.




```
cd /tmp

aws s3 cp s3://iotanalytics-notebook-containers/iota_notebook_containers.zip /tmp

unzip iota_notebook_containers.zip

cd iota_notebook_containers

chmod u+x install.sh

./install.sh
```

Aguarde um ou dois minutos para que a extensão seja validada e instalada.

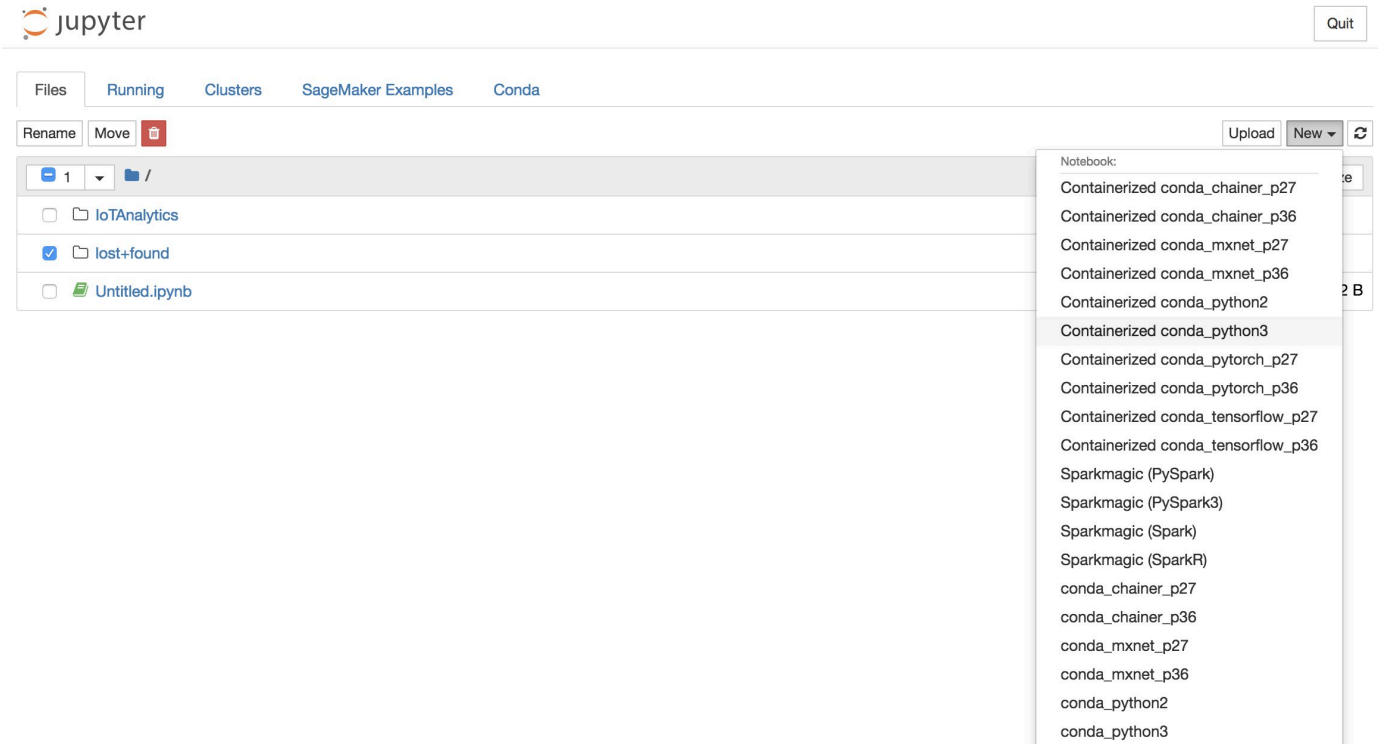
Atualizar a extensão de conteinerização do notebook

Se você tiver criado uma instância do SageMaker por meio do console do AWS IoT Analytics depois de 23 de agosto de 2018, a extensão da conteinerização terá sido instalada automaticamente. Você pode atualizar a extensão reiniciando a instância no console do SageMaker. Se instalou a extensão manualmente, será possível atualizá-la, executando novamente os comandos do terminal listados em Habilitar a conteinerização de instâncias de bloco de anotações não criadas pelo console do AWS IoT Analytics.

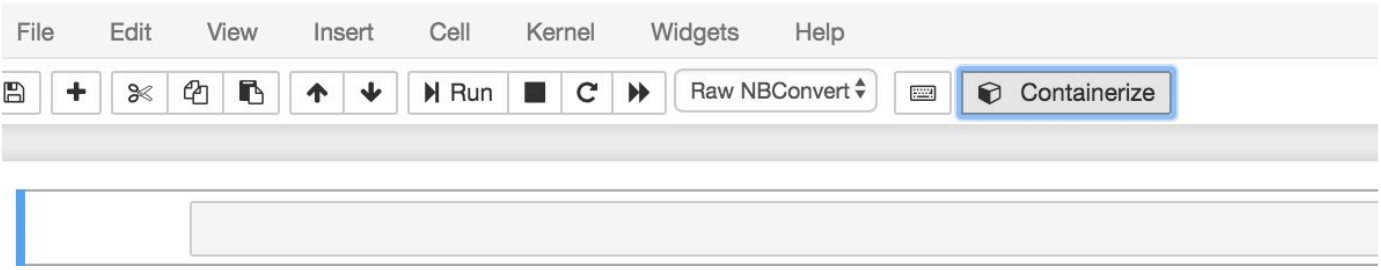
Criar uma imagem conteinerizada

Nesta seção, mostramos as etapas necessárias para conteinerizar um notebook. Para começar, acesse o notebook Jupyter para criar um notebook com um kernel conteinerizado.

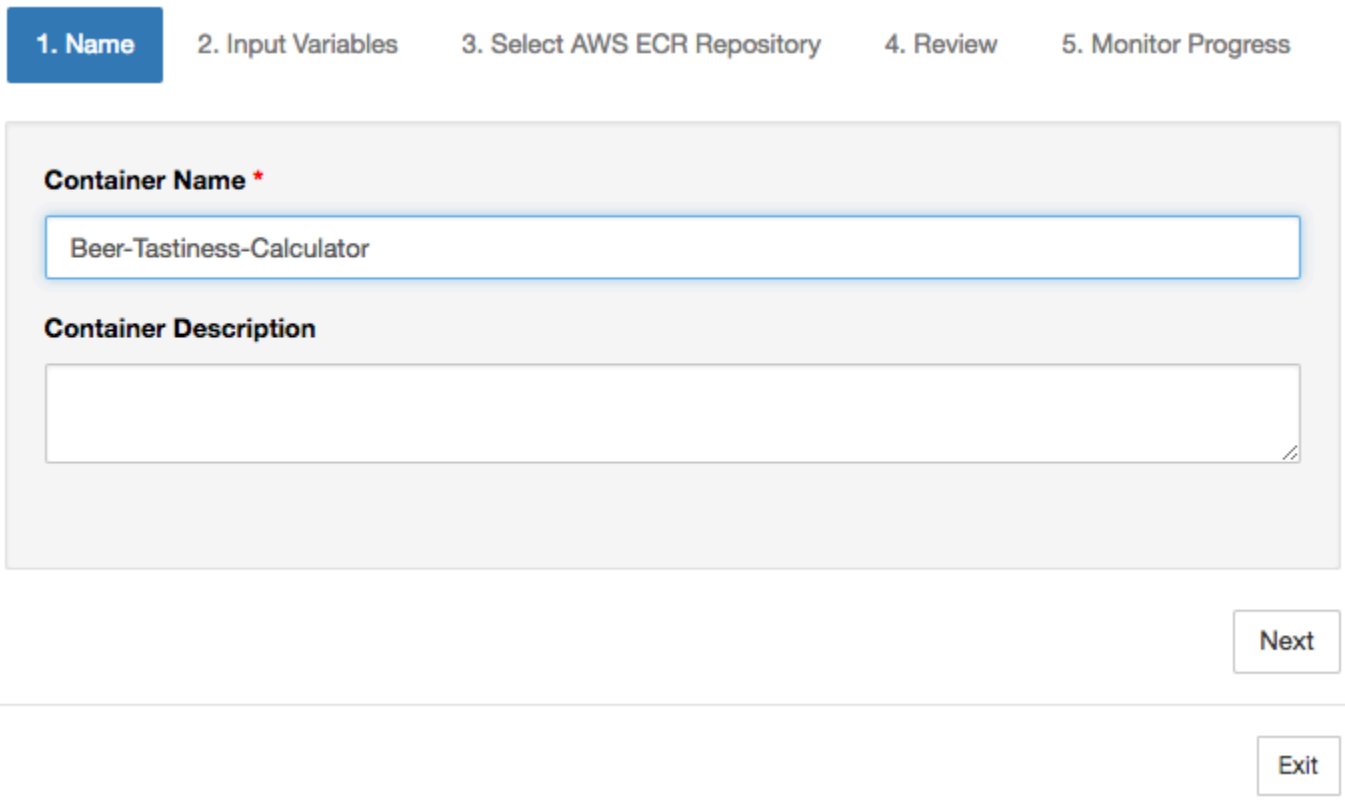
1. No notebook Jupyter, escolha New (Novo) e, em seguida, escolha o tipo de kernel desejado na lista suspensa. (O tipo de kernel deve começar com “Containerized” e terminar com qualquer kernel que você teria selecionado de outra forma. Por exemplo, se você quiser apenas um ambiente Python 3.0 simples, como “conda_python3", escolha “Containerized conda_python3”).



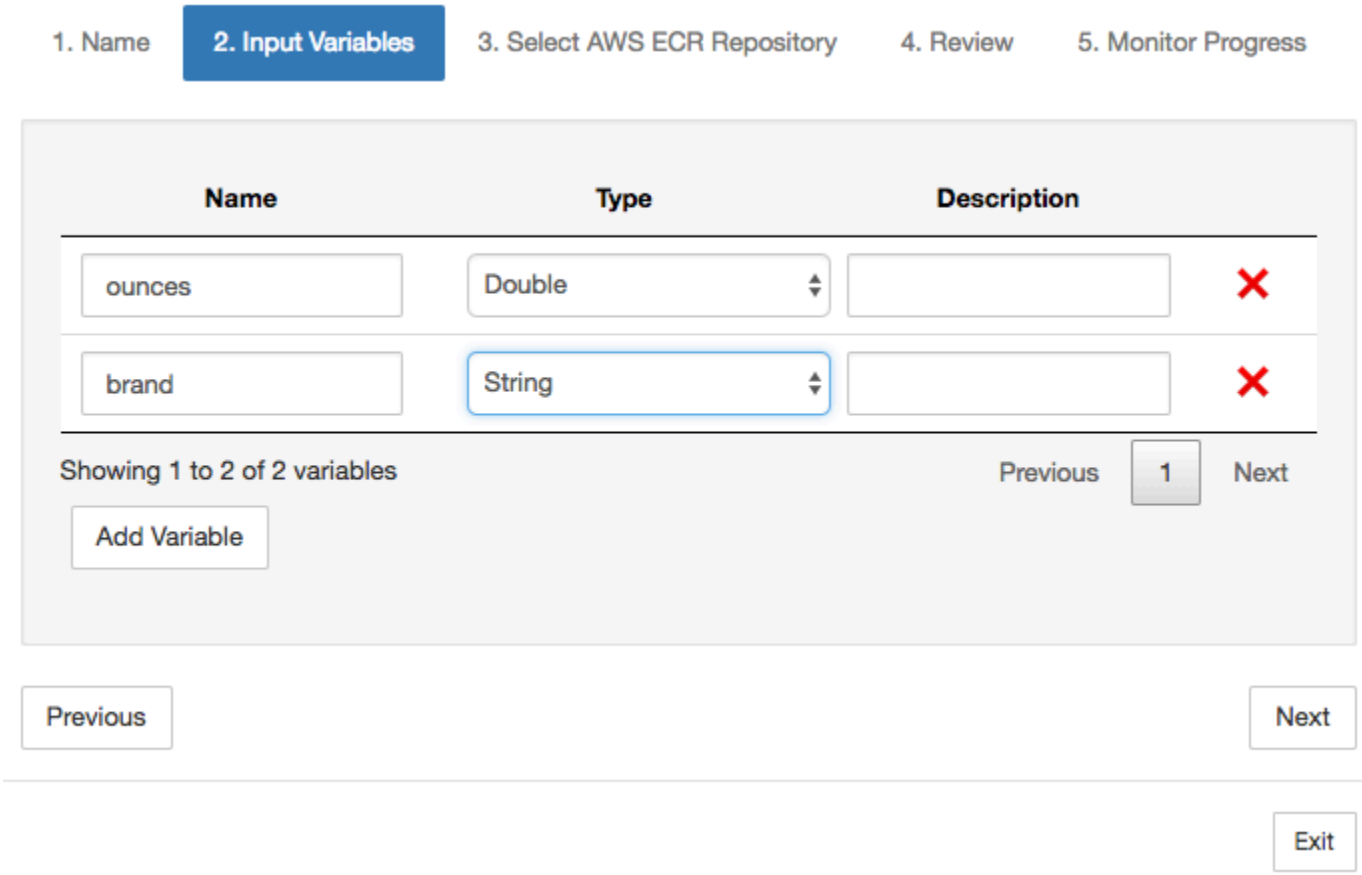
2. Depois de concluir o trabalho no caderno e desejar conteinerizá-lo, escolha o botão Conteinerizar.



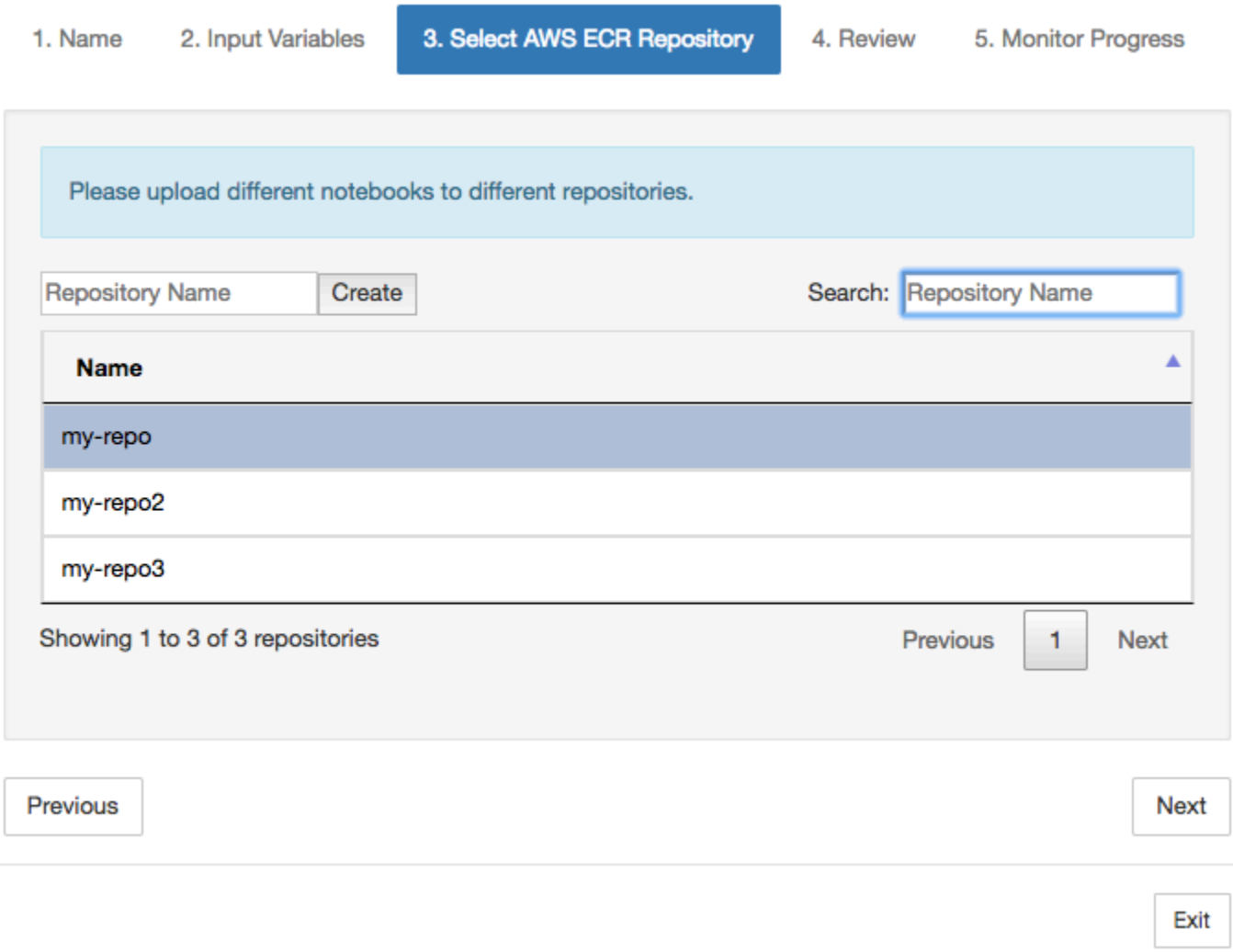
3. Digite um nome para o notebook conteinerizado. Você também pode inserir uma descrição opcional.



4. Especifique as Input Variables (Variáveis de entrada) (parâmetros) com as quais o notebook deve ser invocado. Você pode selecionar as variáveis de entrada que são automaticamente detectadas pelo notebook ou definir variáveis personalizadas. (Observe que as variáveis de entrada só serão detectadas se você já tiver executado o notebook anteriormente.) Para cada variável de entrada, escolha um tipo. Você também pode inserir uma descrição opcional da variável de entrada:

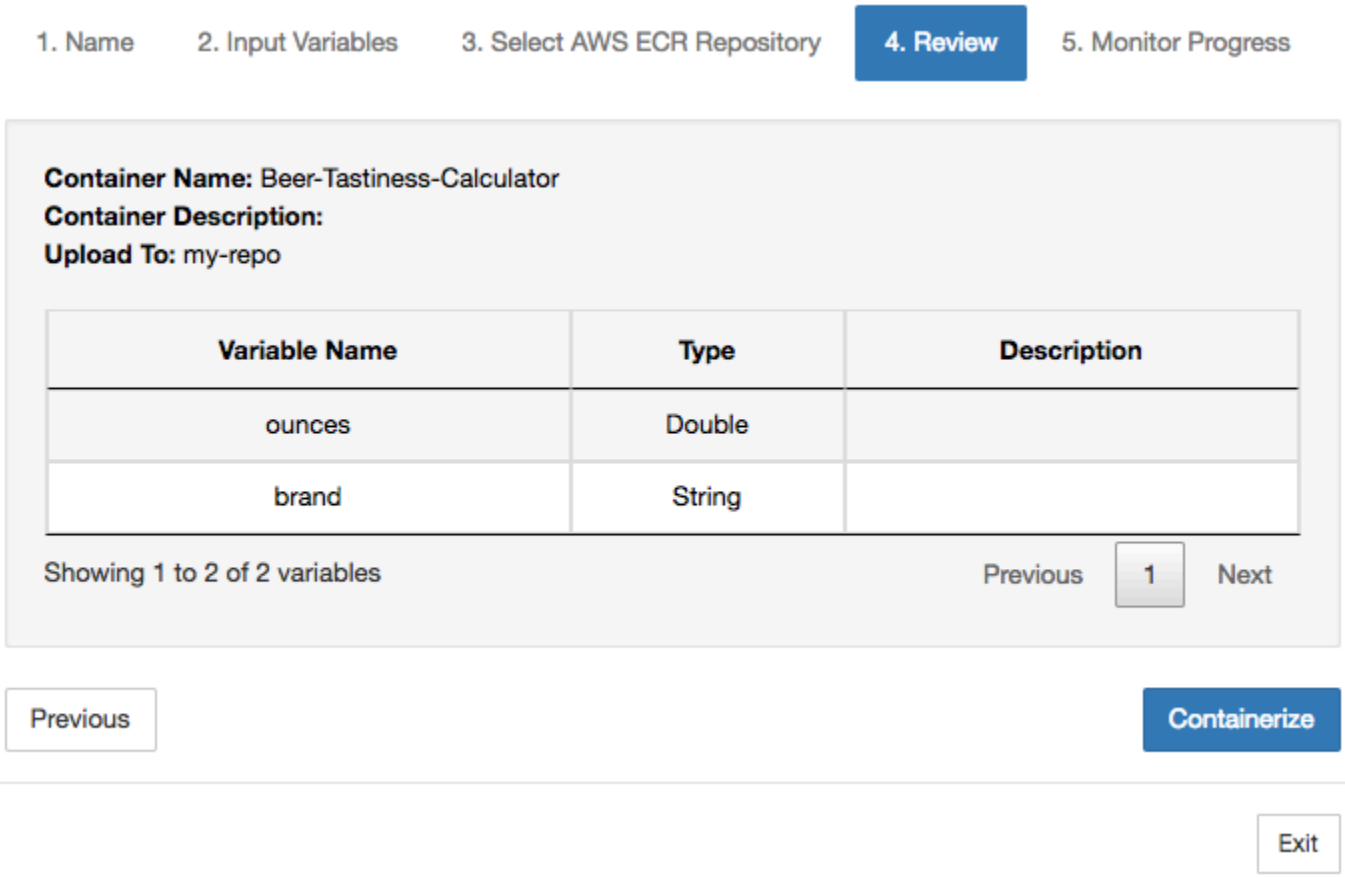


5. Escolha o repositório do Amazon ECR onde a imagem criada do caderno deve ser carregada.

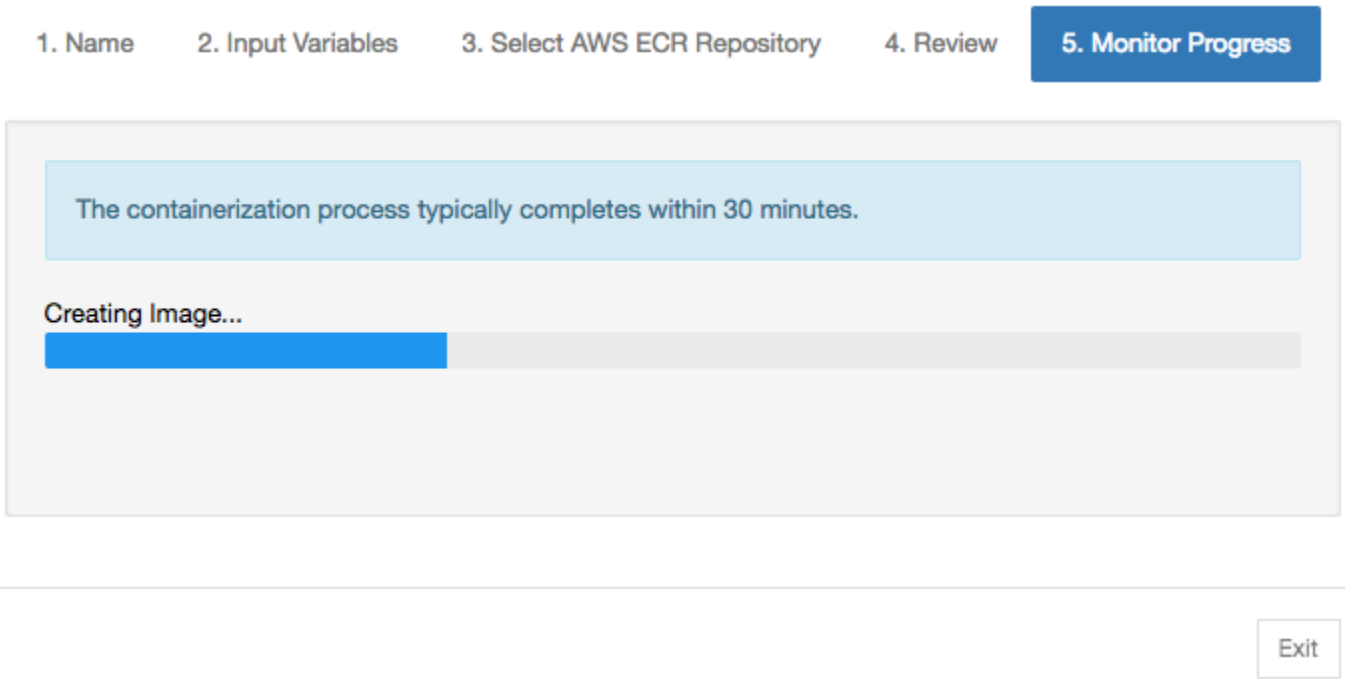


6. Escolha Conteinerizar para começar o processo.

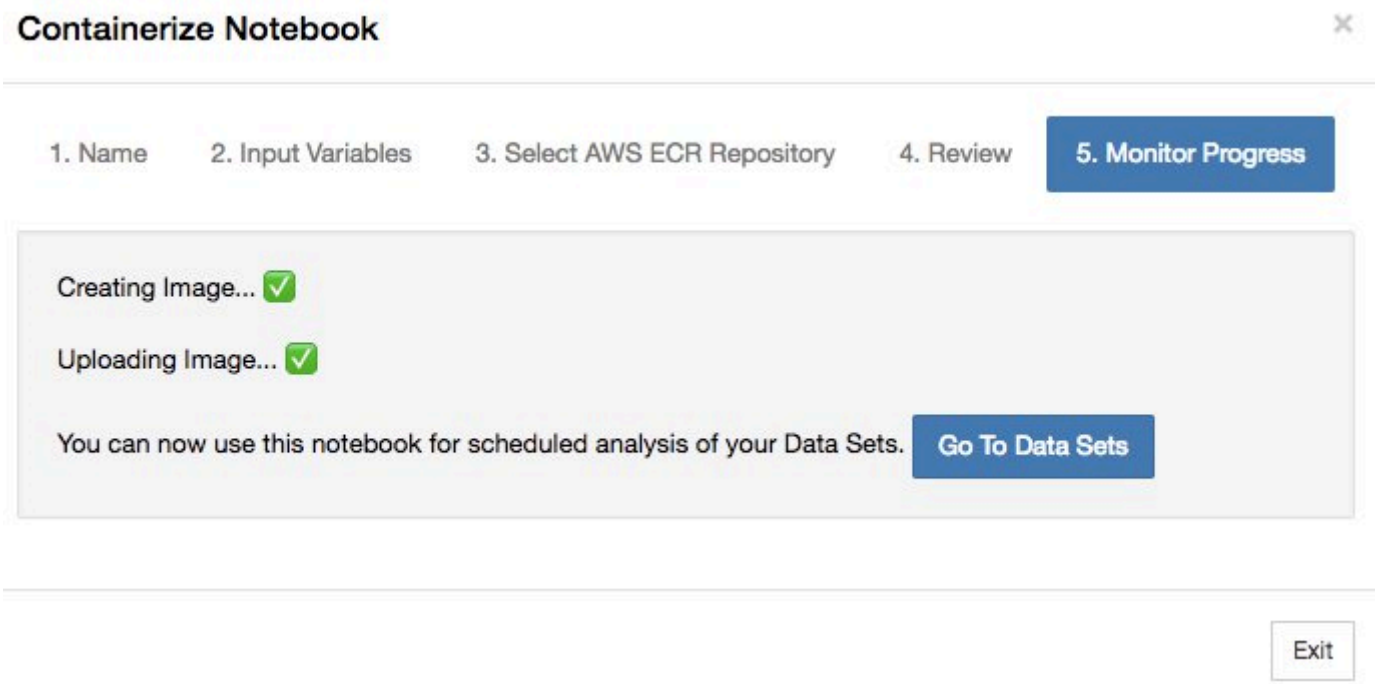
   Será apresentada uma visão geral resumindo sua entrada. Observe que, depois de iniciar o processo, não é possível cancelá-lo. O processo pode durar até uma hora.



7. A próxima página mostra o progresso.



8. Se você fechar o navegador acidentalmente, poderá monitorar o status do processo de conteinerização na seção Cadernos do console do AWS IoT Analytics.
9. Depois que o processo for concluído, a imagem conteinerizada é armazenada no Amazon ECR pronta para uso.



Usando um contêiner personalizado para análise

Esta seção inclui informações sobre como criar um contêiner do Docker usando um caderno Jupyter. Há um risco de segurança se você usar blocos de anotações criados por terceiros: os contêineres incluídos poderão executar código arbitrário com as permissões de usuário. Além disso, o HTML gerado pelo bloco de anotações pode ser exibido em um console do AWS IoT Analytics, fornecendo um possível vetor de ataque no computador que está exibindo o HTML. Certifique-se de confiar no autor de qualquer bloco de anotações de terceiros antes de usá-lo.

É possível criar seu próprio contêiner personalizado e executá-lo com o serviço AWS IoT Analytics. Para fazer isso, você configura e faz upload de uma imagem do Docker no Amazon ECR e, em seguida, configura um conjunto de dados para executar uma ação de contêiner. Esta seção fornece um exemplo do processo usando o Octave.

Este tutorial também pressupõe que você tem:

- o Octave instalado no computador local

- Uma conta de docker configurada no computador local
- Uma conta AWS com Amazon ECR ou acesso AWS IoT Analytics

Etapa 1: Configurar uma imagem de docker

Há três arquivos principais dos quais você precisa para este tutorial. Seus nomes e conteúdo são:

- `Dockerfile` — a configuração inicial do processo de conteinerização do Docker.

```
FROM ubuntu:16.04

# Get required set of software
RUN apt-get update
RUN apt-get install -y software-properties-common
RUN apt-get install -y octave
RUN apt-get install -y python3-pip

# Get boto3 for S3 and other libraries
RUN pip3 install --upgrade pip
RUN pip3 install boto3
RUN pip3 install urllib3

# Move scripts over
ADD moment moment
ADD run-octave.py run-octave.py

# Start python script
ENTRYPOINT ["python3", "run-octave.py"]
```

- `run-octave.py` — analisa o JSON do AWS IoT Analytics, executa o script do Octave e faz upload de artefatos no Amazon S3.

```
import boto3
import json
import os
import sys
from urllib.parse import urlparse

# Parse the JSON from IoT Analytics
with open('/opt/ml/input/data/iotanalytics/params') as params_file:
    params = json.load(params_file)
```

```
variables = params['Variables']

order = variables['order']
input_s3_bucket = variables['inputDataS3BucketName']
input_s3_key = variables['inputDataS3Key']
output_s3_uri = variables['octaveResultS3URI']

local_input_filename = "input.txt"
local_output_filename = "output.mat"

# Pull input data from S3...
s3 = boto3.resource('s3')
s3.Bucket(input_s3_bucket).download_file(input_s3_key, local_input_filename)

# Run Octave Script
os.system("octave moment {} {} {}".format(local_input_filename,
 local_output_filename, order))

# # Upload the artifacts to S3
output_s3_url = urlparse(output_s3_uri)
output_s3_bucket = output_s3_url.netloc
output_s3_key = output_s3_url.path[1:]

s3.Object(output_s3_bucket, output_s3_key).put(Body=open(local_output_filename,
 'rb'), ACL='bucket-owner-full-control')
```

- `moment` — um script do Octave simples que calcula o momento com base em um arquivo de entrada ou saída e uma ordem especificada.

```
#!/usr/bin/octave -qf

arg_list = argv ();
input_filename = arg_list{1};
output_filename = arg_list{2};
order = str2num(arg_list{3});

[D,delimiterOut]=importdata(input_filename)
M = moment(D, order)

save(output_filename,'M')
```

1. Faça download do conteúdo de cada arquivo. Crie um novo diretório e coloque todos os arquivos nele. Em seguida, `cd` para aquele diretório.
2. Execute o comando a seguir.

```
docker build -t octave-moment .
```

3. Você deve ver uma nova imagem no repositório de docker. Instale-o executando o seguinte comando.

```
docker image ls | grep octave-moment
```

Etapa 2: Fazer upload da imagem do Docker em um repositório do Amazon ECR

1. Crie um repositório do Amazon ECR.

```
aws ecr create-repository --repository-name octave-moment
```

2. Obtenha o login para o ambiente do Docker.

```
aws ecr get-login
```

3. Copie a saída e execute-a. A saída deve parecer com algo semelhante ao seguinte:

```
docker login -u AWS -p password -e none https://your-aws-account-
id.dkr.ecr..amazonaws.com
```

4. Marque a imagem criada com a tag do repositório do Amazon ECR.

```
docker tag your-image-id  your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-
moment
```

5. Envie a imagem para o Amazon ECR.

```
docker push your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-moment
```

Etapa 3: Fazer upload dos dados de exemplo em um bucket do Amazon S3

1. Fazer download do seguinte no arquivo `input.txt`.

```
0.857549  -0.987565  -0.467288  -0.252233  -2.298007
 0.030077  -1.243324  -0.692745   0.563276   0.772901
-0.508862  -0.404303  -1.363477  -1.812281  -0.296744
-0.203897   0.746533   0.048276   0.075284   0.125395
 0.829358   1.246402  -1.310275  -2.737117   0.024629
 1.206120   0.895101   1.075549   1.897416   1.383577
```

2. Crie um bucket do Amazon S3 chamado `octave-sample-data-`*`your-aws-account-id`*.
3. Faça upload do arquivo `input.txt` no bucket do Amazon S3 recém-criado. Agora você deve ter um bucket chamado `octave-sample-data-`*`your-aws-account-id`* que contém o arquivo `input.txt`.

Etapa 4: Criar uma função de execução de contêiner

1. Copie o seguinte para um arquivo denominado `role1.json`. Substitua *`your-aws-account-id`* pelo ID da sua conta AWS e *`aws-region`* pela região da AWS dos seus recursos AWS.

> **Note**
>
> Este exemplo inclui uma chave de contexto de condição global para proteger contra o problema de segurança substituto confuso. Para obter mais informações, consulte the section called “Prevenção contra o ataque “Confused deputy” em todos os serviços”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com",
                    "iotanalytics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
              "StringEquals": {
                "aws:SourceAccount": "your-aws-account-id"
```

```
                },
                "ArnLike": {
                  "aws:SourceArn": "arn:aws:iotanalytics:aws-region:your-aws-account-
id:dataset/DOC-EXAMPLE-DATASET"
                }
              }
    ]
}
```

2. Crie uma função que dê permissões de acesso ao SageMaker e AWS IoT Analytics, usando o arquivo `role1.json` que você baixou.

```
aws iam create-role --role-name container-execution-role --assume-role-policy-
document file://role1.json
```

3. Faça o download do seguinte em um arquivo chamado `policy1.json` e substitua *`your-account-id`* pelo ID da sua conta (veja o segundo ARN abaixo `Statement:Resource`).

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Effect": "Allow",
     "Action": [
       "s3:GetBucketLocation",
       "s3:PutObject",
       "s3:GetObject",
       "s3:PutObjectAcl"
     ],
     "Resource": [
       "arn:aws:s3:::*-dataset-*/*",
       "arn:aws:s3:::octave-sample-data-your-account-id/*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "iotanalytics:*"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
```

```
        "ecr:GetAuthorizationToken",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets"
      ],
      "Resource" : "*"
    }
  ]
}
```

4. Crie uma política do IAM, usando o arquivo `policy.json` obtido por download.

```
aws iam create-policy --policy-name ContainerExecutionPolicy --policy-document
 file://policy1.json
```

5. Anexe a política à função.

```
aws iam attach-role-policy --role-name container-execution-role --policy-arn
 arn:aws:iam::your-account-id:policy/ContainerExecutionPolicy
```

Etapa 5: Criar um conjunto de dados com uma ação de contêiner

1. Faça o download do seguinte em um arquivo chamado `cli-input.json` e substitua todas as instâncias de *`your-account-id`* e *`region`* pelos valores apropriados.

```
{
    "datasetName": "octave_dataset",
    "actions": [
```

```
        {
            "actionName": "octave",
            "containerAction": {
                "image": "your-account-id.dkr.ecr.region.amazonaws.com/octave-
moment",
                "executionRoleArn": "arn:aws:iam::your-account-id:role/container-
execution-role",
                "resourceConfiguration": {
                    "computeType": "ACU_1",
                    "volumeSizeInGB": 1
                },
                "variables": [
                    {
                        "name": "octaveResultS3URI",
                        "outputFileUriValue": {
                            "fileName": "output.mat"
                        }
                    },
                    {
                        "name": "inputDataS3BucketName",
                        "stringValue": "octave-sample-data-your-account-id"
                    },
                    {
                        "name": "inputDataS3Key",
                        "stringValue": "input.txt"
                    },
                    {
                        "name": "order",
                        "stringValue": "3"
                    }
                ]
            }
        }
    ]
}
```

2. Crie um conjunto de dados usando o arquivo `cli-input.json` obtido por download e editado.

```
aws iotanalytics create-dataset –cli-input-json file://cli-input.json
```

Etapa 6: Invocar a geração do conteúdo do conjunto de dados

1. Execute o comando a seguir.

```
aws iotanalytics create-dataset-content --dataset-name octave-dataset
```

Etapa 7: Obter o conteúdo do conjunto de dados

1. Execute o comando a seguir.

```
aws iotanalytics get-dataset-content --dataset-name octave-dataset --version-id \
$LATEST
```

2. É necessário esperar alguns minutos até que o `DatasetContentState` seja `SUCCEEDED`.

Etapa 8: Imprimir a saída no Octave

1. Use o shell do Octave para imprimir a saída do contêiner executando o seguinte comando:

```
bash> octave
octave> load output.mat
octave> disp(M)
-0.016393 -0.098061 0.380311 -0.564377 -1.318744
```

Visualizando dados AWS IoT Analytics

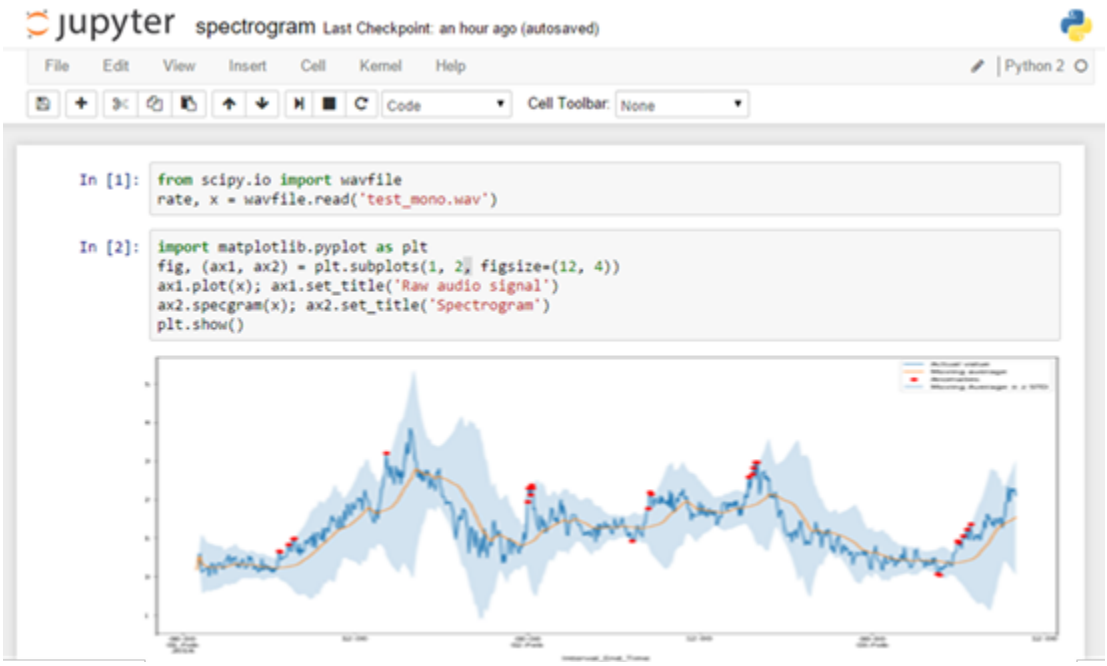
Para visualizar seus dados de AWS IoT Analytics, você pode usar o console AWS IoT Analyticsou o Amazon QuickSight.

Tópicos

- Visualizando dados AWS IoT Analytics com o console
- Visualização de dados AWS IoT Analytics com o Amazon QuickSight

Visualizando dados AWS IoT Analytics com o console

O AWS IoT Analytics pode incorporar a saída HTML do conjunto de dados de contêiner (encontrada no arquivo `output.html`) na página de conteúdo do conjunto de dados de contêiner do AWS IoT Analytics console. Por exemplo, se você definir um conjunto de dados de contêiner que executa um caderno Jupyter e criar uma visualização no caderno Jupyter, seu conjunto de dados pode ser semelhante ao seguinte:



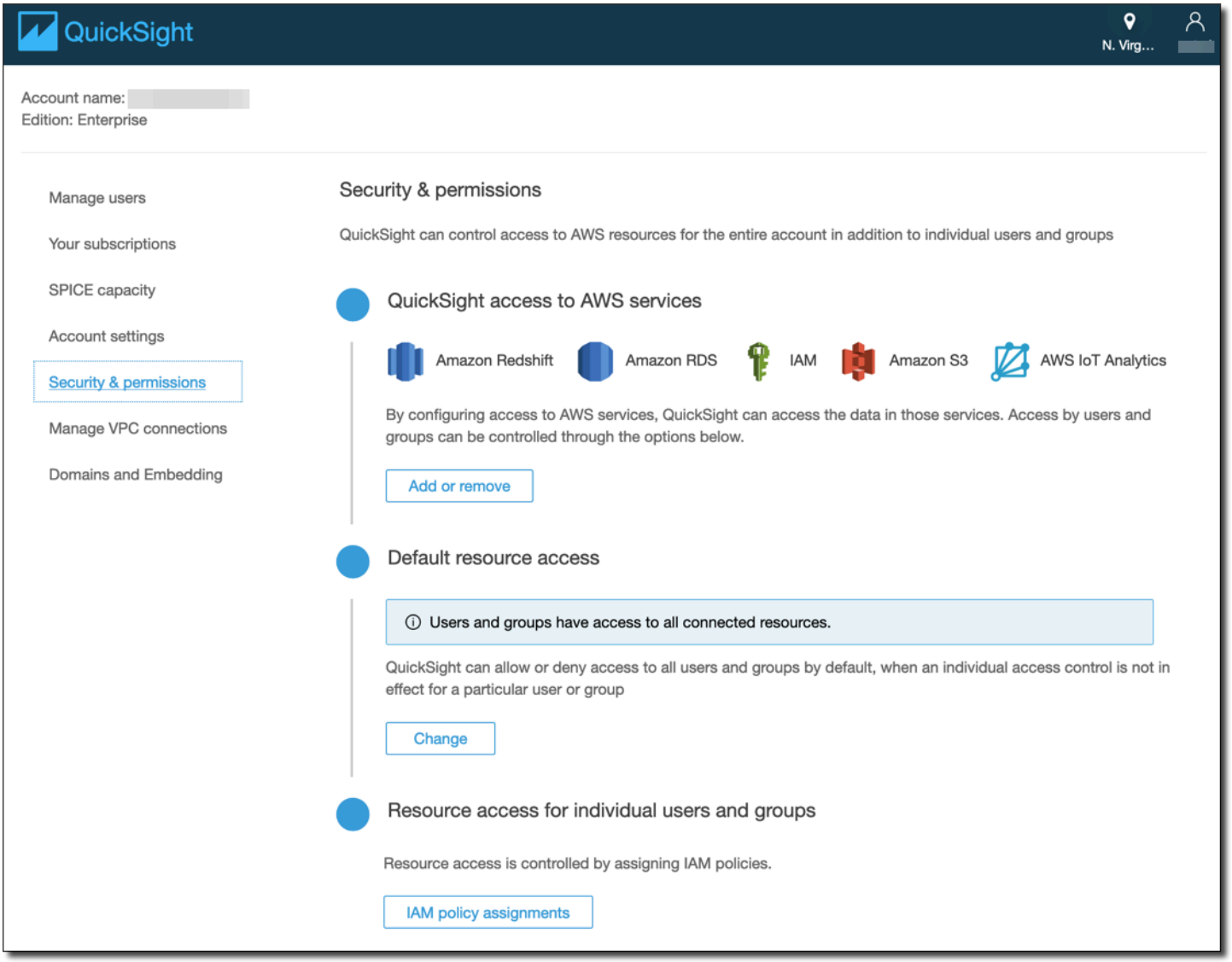
Depois que o conteúdo do conjunto de dados de contêiner for criado, você poderá ter essa visualização na página de conteúdo do Conjunto de dados do console.

Para obter informações sobre como criar um conjunto de dados de contêiner que executa um caderno Jupyter, consulte Automatizar seu fluxo de trabalho.

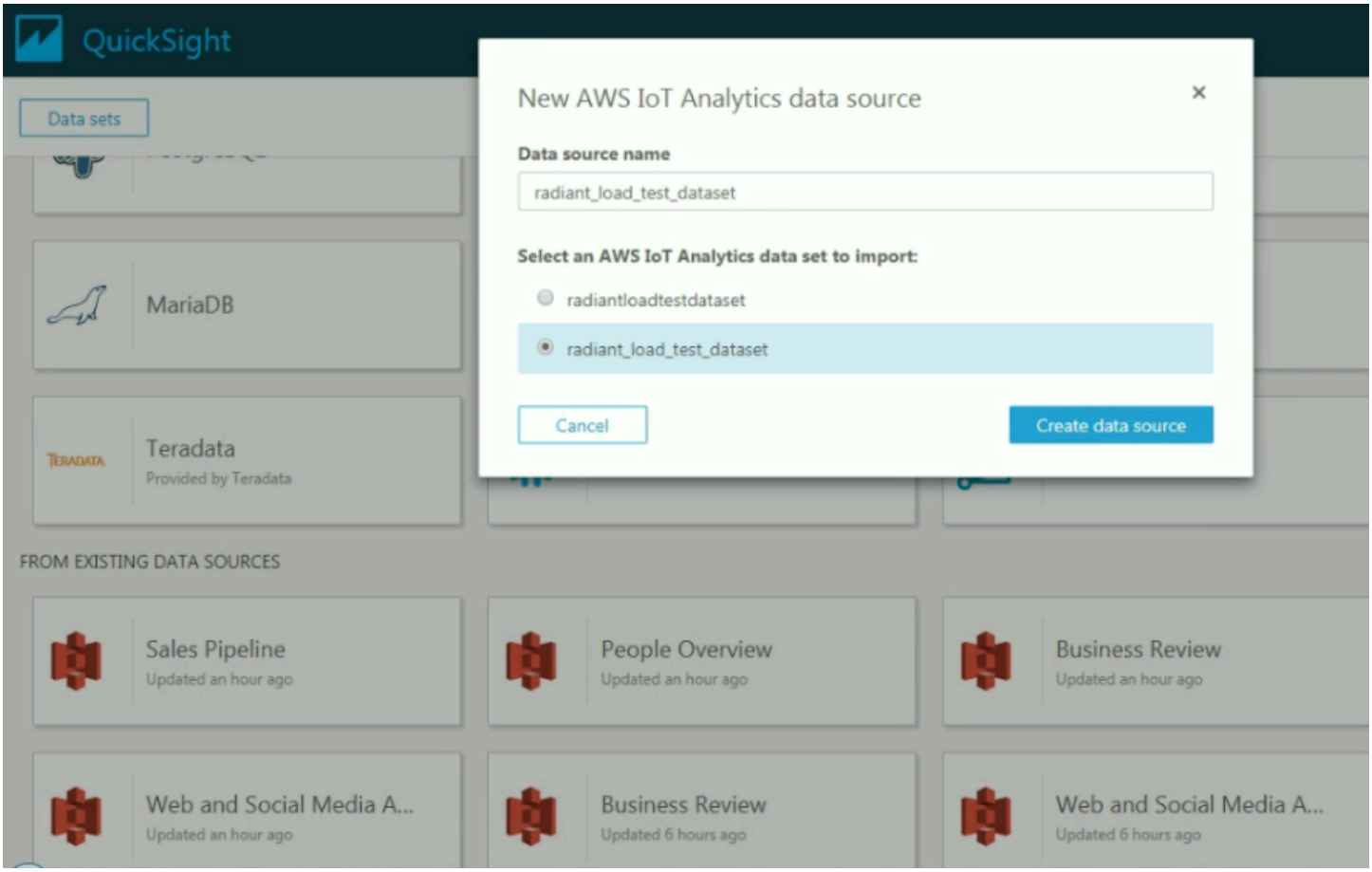
Visualização de dados AWS IoT Analytics com o Amazon QuickSight

O AWS IoT Analytics fornece integração direta com o Amazon QuickSight. O Amazon QuickSight é um serviço de análise de negócios rápido que você pode usar para criar visualizações, realizar uma análise ad hoc e obter insights de seus dados rapidamente. O Amazon QuickSight permite que as organizações escalem para centenas de milhares de usuários e fornece desempenho responsivo usando um mecanismo de memória robusto (SPICE). Você pode selecionar seus conjuntos de dados do AWS IoT Analytics no console do Amazon QuickSight e começar a criar painéis e visualizações. O Amazon QuickSight está disponível nestas regiões.

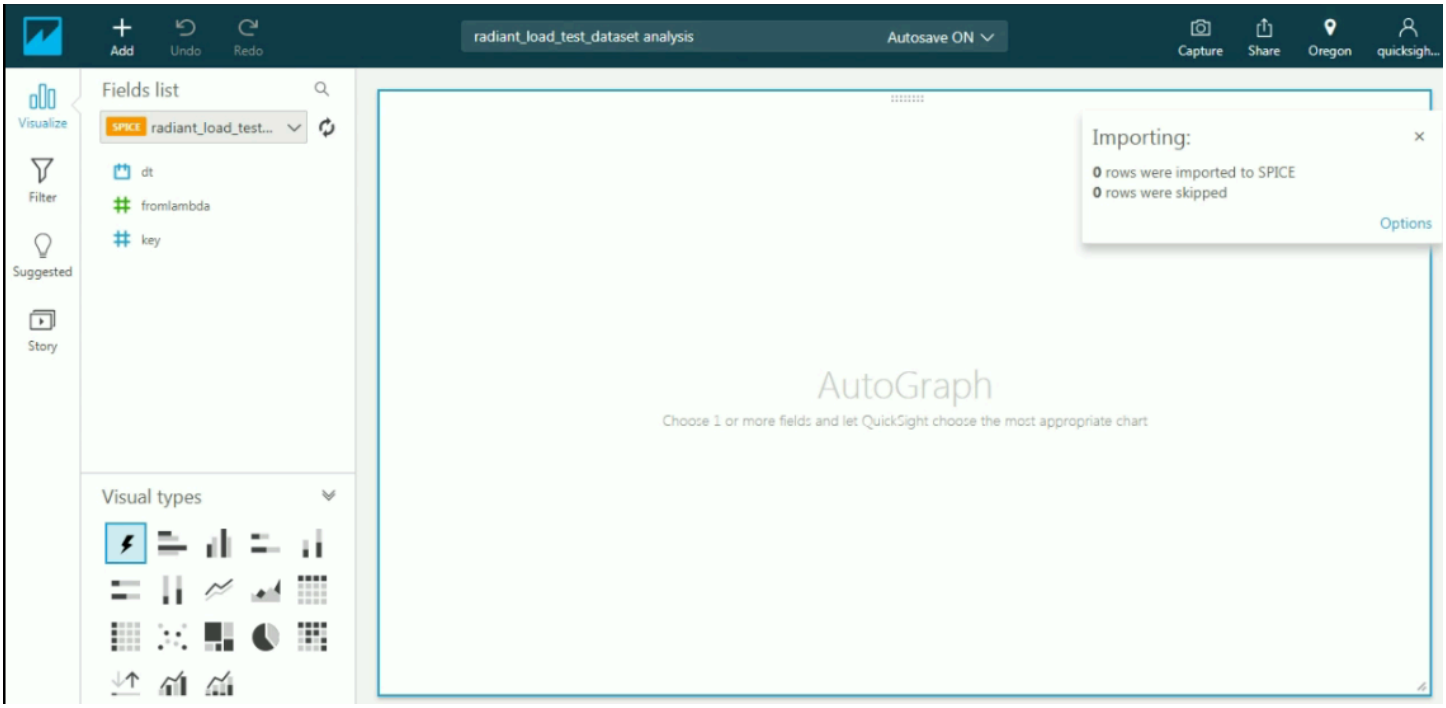
Para começar a usar as visualizações do Amazon QuickSight, é necessário criar uma conta do Amazon QuickSight. Certifique-se de conceder ao Amazon QuickSight acesso aos dados do AWS IoT Analytics ao configurar sua conta. Se você já tiver uma conta, dê ao Amazon QuickSight acesso aos seus dados do AWS IoT Analytics escolhendo Admin, Gerenciar QuickSight, Segurança e permissões. Em Acesso QuickSight aos serviços da AWS, escolha Adicionar ou remover e, em seguida, marque a caixa de seleção ao lado de AWS IoT Analytics e escolha Atualizar.



Depois que sua conta estiver configurada, na página do console do admin do Amazon QuickSight, escolha Nova análise e Novo conjunto de dados e, em seguida, selecione AWS IoT Analytics como a fonte. Digite um nome para a fonte de dados, selecione um conjunto de dados para importar e, em seguida, selecione Criar fonte de dados.



Depois que a fonte de dados tiver sido criada, você poderá criar visualizações no Amazon QuickSight.



Para obter informações sobre painéis e conjuntos de dados do Amazon QuickSight, consulte a Documentação do Amazon QuickSight.

Marcar recursos do AWS IoT Analytics

Para ajudar a gerenciar seus canais, conjuntos de dados, datastores e pipelines, você pode atribuir seus próprios metadados a cada um desses recursos na forma de tags. Este capítulo descreve as tags e mostra como criá-las.

Tópicos

- Conceitos básicos de tags
- Utilização de tags com políticas do IAM
- Restrições de tags

Conceitos básicos de tags

As tags permitem categorizar seus recursos da AWS IoT Analytics de diferentes formas (como por finalidade, por proprietário ou por ambiente). Isso é útil quando há muitos recursos do mesmo tipo — você pode identificar rapidamente um recurso específico com base nas tags que atribuiu a ele. Cada tag consiste em uma chave e em um valor opcional, ambos definidos por você. Por exemplo, você pode definir um conjunto de tags para os canais que ajude a rastrear o tipo de dispositivo responsável por cada origem de mensagem do canal. Recomendamos que você desenvolva um conjunto de chave de tags que atenda suas necessidades para cada tipo de recurso. Usar um conjunto consistente de chaves de tags facilita para você gerenciar seus recursos. É possível pesquisar e filtrar os recursos de acordo com as tags que adicionar.

Você também pode usar tags para categorizar e rastrear seus custos. Quando você aplica tags a canais, conjuntos de dados, datastores ou pipelines, a AWS gera um relatório de alocação de custos como um arquivo CSV (valores separados por vírgula) com o uso e os custos agregados por suas tags. É possível aplicar tags que representem categorias de negócios (como centros de custos, nomes de aplicativos ou proprietários) para organizar seus custos de vários serviços. Para obter mais informações sobre como usar tags para alocação de custos, consulte Usar tags de alocação de custos no Guia do usuário AWS Billing.

Para facilidade de uso, use o Editor de tags no console AWS Billing and Cost Management, que fornece uma maneira unificada e central para criar e gerenciar suas tags. Para obter mais informações, consulte Como trabalhar com o Tag Editor em Conceitos básicos do AWS Management Console.

Você também pode trabalhar com tags usando a AWS CLI e a API do AWS IoT Analytics. Você pode associar tags a canais, conjuntos de dados, datastores e pipelines ao criá-los. Use o campo Tags nos seguintes comandos:

- CreateChannel
- CreateDataset
- CreateDatastore
- CreatePipeline

É possível adicionar, modificar ou excluir tags de recursos existentes que oferecem suporte a marcação. Use os seguintes comandos:

- TagResource
- ListTagsForResource
- UntagResource

É possível editar chaves de tags e valores, e é possível remover as tags de um recurso a qualquer momento. É possível definir o valor de uma tag a uma string vazia, mas não pode configurar o valor de um tag como nula. Se você adicionar uma tag que tenha a mesma chave de uma tag existente nesse recurso, o novo valor substituirá o antigo. Se você excluir um recurso, todas as tags associadas ao recurso também serão excluídas.

Utilização de tags com políticas do IAM

É possível usar o elemento `Condition` (também chamado bloco `Condition`) com os seguintes valores e chaves de contexto de condição em uma política do IAM para controlar o acesso do usuário (permissões) com base nas tags de um recurso:

- Use `iotanalytics:ResourceTag/<tag-key>: <tag-value>` para permitir ou negar ações do usuário em recursos com tags específicas.
- Use `aws:RequestTag/<tag-key>: <tag-value>` para exigir que uma tag específica seja (ou não seja) usada ao fazer uma solicitação de API para criar ou modificar um recurso que permite tags.
- Use `aws:TagKeys: [<tag-key>, ...]` para exigir que um conjunto específico de chaves de tag seja (ou não seja) usado ao fazer uma solicitação de API para criar ou modificar um recurso que permite tags.

> **Note**
>
> Os valores/chaves de contexto de condição em uma política do IAM se aplicam somente às ações do AWS IoT Analytics em que um identificador de um recurso que pode ser marcado com tags é um parâmetro obrigatório. Por exemplo, o uso de DescribeLoggingOptions não é permitido/negado com base em valores e chaves de contexto de condição, pois nenhum recurso que pode ser marcado (canal, conjunto de dados, datastore ou pipeline) é referenciado nesta solicitação.

Para mais informações, consulte Controlar o acesso usando etiquetas no Guia do usuário do IAM. A seção Referência de política JSON do IAM desse guia detalhou a sintaxe, as descrições e os exemplos dos elementos, variáveis e lógica de avaliação das políticas JSON no IAM.

A política de exemplo a seguir aplica duas restrições com base em tag. Um usuário restrito por essa política:

1. não pode atribuir um recurso à tag “env=prod” (consulte a linha `"aws:RequestTag/env" : "prod"` no exemplo).
2. não pode modificar ou acessar um recurso que tenha uma tag “env=prod” existente (consulte a linha `"iotanalytics:ResourceTag/env" : "prod"` no exemplo).

```
{
  "Version" : "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "aws:RequestTag/env" : "prod"
        }
      }
    },
    {
      "Effect" : "Deny",
```

```
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "iotanalytics:ResourceTag/env" : "prod"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Você também pode especificar vários valores de tag para uma determinada chave de tag, colocando-as em uma lista como o exemplo a seguir:

```
"StringEquals" : {
  "iotanalytics:ResourceTag/env" : ["dev", "test"]
}
```

> **Note**
>
> Se você permitir ou negar aos usuários o acesso a recursos com base em tags, é importante considerar negar explicitamente aos usuários a capacidade de adicionar essas tags ou removê-las dos mesmos recursos. Caso contrário, é possível que um usuário contorne suas restrições e obtenha acesso a um recurso modificando as tags.

Restrições de tags

As restrições básicas a seguir se aplicam às tags:

- Número máximo de tags por recurso — 50
- Comprimento máximo da chave — 127 caracteres Unicode em UTF-8
- Valor máximo da chave — 255 caracteres Unicode em UTF-8

- As chaves e os valores de tags diferenciam maiúsculas de minúsculas.
- Não use o `aws: prefix` no nome nem no valor de suas tags, pois ele é reservado para uso da AWS. Você não pode editar nem excluir nomes ou valores de tag com esse prefixo. As tags com esse prefixo não contam para as tags por limite de origem.
- Se seu esquema de tags é usado em vários serviços e recursos, lembre-se de que outros serviços podem ter restrições nos caracteres permitidos. Em geral, os caracteres permitidos são letras, espaços e números representáveis em UTF-8, além dos seguintes caracteres especiais: + - = . _ : / @.

Expressões SQL em AWS IoT Analytics

Os conjuntos de dados são gerados usando expressões SQL em dados em um datastore. AWS IoT Analytics usa as mesmas consultas SQL, funções e operadores do Amazon Athena.

AWS IoT Analytics oferece compatibilidade com um subconjunto da sintaxe SQL padrão ANSI.

```
SELECT [ ALL | DISTINCT ] select_expression [, ...]
[ FROM from_item [, ...] ]
[[ INNER | OUTER ] LEFT | RIGHT | FULL | CROSS JOIN join_item [ ON join_condition ]]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition ]
[ UNION [ ALL | DISTINCT ] union_query ]
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]
[ LIMIT [ count | ALL ] ]
```

Para obter uma descrição dos parâmetros, consulte Parâmetros na documentação do Amazon Athena.

AWS IoT Analytics e o Amazon Athena não é compatível com o seguinte:

- Cláusulas `WITH`
- Instruções `CREATE TABLE AS SELECT`
- Instruções `INSERT INTO`
- Instruções preparadas, você não pode executar `EXECUTE` com `USING`.
- `CREATE TABLE LIKE`
- `DESCRIBE INPUT` e `DESCRIBE OUTPUT`
- Instruções `EXPLAIN`
- User-Defined Functions (UDFs – Funções definidas pelo usuário ou UDAFs)
- Procedimentos armazenados
- Conectores federados

Tópicos

- Funcionalidade SQL compatível com AWS IoT Analytics

- Solucionar problemas comuns com consultas SQL no AWS IoT Analytics

Funcionalidade SQL compatível com AWS IoT Analytics

Os conjuntos de dados são gerados por meio de expressões SQL em dados em um datastore. As consultas que você executa no AWS IoT Analytics são baseadas no Presto 0.217.

Tipos de dados compatíveis

AWS IoT Analytics e o Amazon Athena são compatíveis com esses tipos de dados.

- primitive_type
  - `TINYINT`
  - `SMALLINT`
  - `INT`
  - `BIGINT`
  - `BOOLEAN`
  - `DOUBLE`
  - `FLOAT`
  - `STRING`
  - `TIMESTAMP`
  - `DECIMAL(precision, scale)`
  - `DATE`
  - `CHAR` (dados de caractere de comprimento fixo com um tamanho especificado)
  - `VARCHAR` (dados de caractere de comprimento variável com um tamanho especificado)
- array_type
  - `ARRAY<data_type>`
- map_type
  - `MAP<primitive_type, data_type>`
- struct_type
  - `STRUCT<col_name:data_type[COMMENT col_comment][,...]>`

> **Note**
>
> AWS IoT Analytics e o Amazon Athena não são compatíveis com alguns tipos de dados.

Funções compatíveis

As funcionalidades do Amazon Athena e AWS IoT Analytics do SQL são baseadas no Presto 0.217. Para obter informações sobre funções, operadores e expressões relacionados, consulte Funções e operadores e as seções a seguir específicas da documentação do Presto.

- Operadores lógicos
- Funções e operadores comparativos
- Expressões condicionais
- Funções de conversão
- Funções e operadores matemáticos
- Funções bitwise
- Funções e operadores decimais
- Funções e operadores de string
- Funções binárias
- Funções e operadores de data e hora
- Funções de expressões regulares
- Funções e operadores JSON
- Funções de URL
- Funções agregadas
- Funções de janela
- Funções de cor
- Funções e operadores de matriz
- Funções e operadores de mapa
- Expressões e funções do Lambda
- Funções de teradados

> **Note**
>
> AWS IoT Analytics e o Amazon Athena não são compatíveis com funções definidas pelo usuário (UDFs ou UDAFs) ou procedimentos armazenados.

Solucionar problemas comuns com consultas SQL no AWS IoT Analytics

Use as informações a seguir para ajudar a solucionar problemas com suas consultas SQL no AWS IoT Analytics.

- Para inserir aspas simples, preceda-as com outras aspas simples. Não confunda isso com aspas duplas.

  Example Exemplo

  ```
  SELECT 'O''Reilly'
  ```

- Para inserir sublinhados, use acentos indicativos de crase para delimitar os nomes de coluna do datastore que comecem com um sublinhado.

  Example Exemplo

  ```
  SELECT `_myMessageAttribute` FROM myDataStore
  ```

- Para inserir nomes com números, delimite os nomes de datastore que incluam números entre aspas duplas.

  Example Exemplo

  ```
  SELECT * FROM "myDataStore123"
  ```

- Para inserir palavras-chave reservadas, delimite as palavras-chave reservadas entre aspas duplas. Para obter mais informações, consulte Lista de palavras-chave reservadas nas instruções SQL SELECT.

Segurança em AWS IoT Analytics

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. A eficácia da nossa segurança é regularmente testada e verificada por auditores de terceiros como parte dos Programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam AWS IoT Analytics, consulte AWS serviços no escopo por programa de conformidade.
- Segurança na nuvem - Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, inclusive a confidencialidade dos dados, os requisitos da organização, as leis e as regulamentações vigentes.

Esta documentação ajudará você a entender como aplicar o modelo de responsabilidade compartilhada ao usar AWS IoT Analytics. Os tópicos a seguir mostram como configurar para atender AWS IoT Analytics aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que podem ajudá-lo a monitorar e proteger seus AWS IoT Analytics recursos.

AWS Identity and Access Management em AWS IoT Analytics

AWS Identity and Access Management (IAM) é um AWS serviço que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar AWS IoT Analytics os recursos. O IAM é um AWS serviço que você pode usar sem custo adicional.

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz AWS IoT Analytics.

Usuário do serviço — Se você usar o AWS IoT Analytics serviço para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais AWS IoT Analytics recursos para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no AWS IoT Analytics, consulte Solução de problemas AWS IoT Analytics de identidade e acesso.

Administrador de serviços — Se você é responsável pelos AWS IoT Analytics recursos da sua empresa, provavelmente tem acesso total AWS IoT Analytics a. É seu trabalho determinar quais AWS IoT Analytics recursos e recursos seus usuários do serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como sua empresa pode usar o IAM com AWS IoT Analytics, consulteComo AWS IoT Analytics funciona com o IAM.

Administrador do IAM: Se você for um administrador do IAM, talvez queira saber detalhes sobre como pode gravar políticas para gerenciar acesso ao AWS IoT Analytics. Para ver exemplos de políticas AWS IoT Analytics baseadas em identidade que você pode usar no IAM, consulte. AWS IoT Analytics exemplos de políticas baseadas em identidade

Autenticando com identidades

A autenticação é 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 assinar solicitações por conta própria, consulte Assinatura de solicitações de AWS API no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser exigido que você forneça 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 AWS IAM Identity Center Guia do Usuário. Usar a autenticação multifator (MFA) na AWS 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 pode 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, recomendamos depender de 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, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte Alterne 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 nome de grupo IAMAdmins e atribuir a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a um aplicativo, mas uma função pode ser assumida por qualquer pessoa que precisar dela. Os usuários

têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando Criar um Usuário do IAM (Ao Invés de uma Função) 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. Você pode assumir temporariamente uma função do IAM no AWS Management Console trocando de funções. 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 Usando Funções do IAM no Guia do Usuário do IAM.

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

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode 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 pelo mesmo. Para obter mais informações sobre perfis para federação, consulte Criando um Perfil para um Provedor de Identidades Terceirizado no Guia do Usuário do IAM. Se você usa o IAM Identity Center, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no AWS IAM Identity Center Manual do Usuário.
- Permissões de usuários temporárias do IAM: um usuário ou perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: você pode usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) acesse recursos na sua conta de uma conta diferente. As funções são a forma primária 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 aprender a diferença entre funções e políticas baseadas em recurso para acesso entre contas, consulte Como as Funções do IAM Diferem das Políticas Baseadas em Recurso 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 no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso

usando as permissões de chamada da entidade principal, uma função de serviço ou uma função vinculada ao serviço.

- Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações em 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 Encaminhar sessões de acesso.
- Função de Serviço: uma função de serviço é uma função do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criando 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 assumir 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 pode editar as permissões para funções vinculadas a serviço.

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

Para aprender se deseja usar perfis do IAM, consulte Quando Criar uma Função do IAM (em Vez de um Usuário) no Guia do Usuário do IAM.

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

As políticas do IAM definem permissões para uma ação, independente 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 do IAM, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em quais condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

As políticas baseadas em identidade também podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são incorporadas diretamente a um único usuário, grupo ou função. 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 selecionar entre uma política gerenciada ou uma política em linha, consulte Selecionar entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

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 e dos seus limites de permissões. As políticas baseadas em atributo 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 Entidades 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 atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as 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 o Organizações e SCPs, consulte Como os SCPs Funcionam no AWS Organizations Manual do Usuário do.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para uma função 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 atributo. 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 AWS IoT Analytics funciona com o IAM

Antes de usar o IAM para gerenciar o acesso AWS IoT Analytics, você deve entender quais recursos do IAM estão disponíveis para uso AWS IoT Analytics. Para ter uma visão de alto nível de como

AWS IoT Analytics e outros AWS serviços funcionam com o IAM, consulte AWS os serviços que funcionam com o IAM no Guia do usuário do IAM.

Tópicos nesta página:

- AWS IoT Analytics políticas baseadas em identidade
- AWS IoT Analytics políticas baseadas em recursos
- Autorização baseada em AWS IoT Analytics tags
- AWS IoT Analytics Funções do IAM

AWS IoT Analytics políticas baseadas em identidade

Com as políticas baseadas em identidade do IAM, você pode especificar ações e recursos permitidos ou negados e as condições sob as quais as ações são permitidas ou negadas. AWS IoT Analytics oferece suporte a ações, recursos e chaves de condição específicos. Para saber mais sobre todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações

O elemento `Action` de uma política baseada em identidade do IAM descreve a ação ou ações específicas que serão permitidas ou negadas pela política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. A ação é usada em uma política para conceder permissões para executar a operação associada.

A ação política AWS IoT Analytics usa o seguinte prefixo antes da ação: Por exemplo, `iotanalytics:` para conceder permissão a alguém para criar um AWS IoT Analytics canal com a operação da AWS IoT Analytics `CreateChannel` API, você inclui a `iotanalytics:BatchPuMessage` ação na política dessa pessoa. As declarações de política devem incluir um `NotAction` elemento `Action` ou. AWS IoT Analytics define seu próprio conjunto de ações que descrevem as tarefas que você pode executar com esse serviço.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, conforme a seguir.

```
"Action": [
    "iotanalytics:action1",
    "iotanalytics:action2"
    ]
```

Você também pode especificar várias ações utilizando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a ação a seguir:

```
"Action": "iotanalytics:Describe*"
```

Para ver uma lista de AWS IoT Analytics ações, consulte Ações definidas AWS IoT Analytics no Guia do usuário do IAM.

Recursos

O elemento `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Você especifica um recurso usando um ARN ou usando o caractere curinga (*) para indicar que a instrução se aplica a todos os recursos.

O recurso do AWS IoT Analytics conjunto de dados tem o seguinte ARN.

```
arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/${DatasetName}
```

Para obter mais informações sobre o formato de ARNs, consulte Nomes de recursos da Amazon (ARNs) e namespaces de serviços da AWS.

Por exemplo, para especificar o conjunto de dados `Foobar` em sua instrução, use o seguinte ARN.

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/Foobar"
```

Para especificar todas as instâncias que pertencem a uma conta específica, use o caractere curinga (*).

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/*"
```

Algumas AWS IoT Analytics ações, como as de criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, você deve utilizar o caractere curinga (*).

```
"Resource": "*"
```

Algumas ações AWS IoT Analytics da API envolvem vários recursos. Por exemplo, `CreatePipeline` faz referência a um canal e a um conjunto de dados, portanto, um usuário deve ter permissões para usar o canal e o conjunto de dados. Para especificar vários recursos em uma única instrução, separe os ARNs com vírgulas.

```
"Resource": [
    "resource1",
    "resource2"
     ]
```

Para ver uma lista dos tipos de AWS IoT Analytics recursos e seus ARNs, consulte Recursos definidos por AWS IoT Analytics no Guia do usuário do IAM. Para saber com quais ações é possível especificar o ARN de cada atributo, consulte Ações definidas pelo AWS IoT Analytics.

Chaves de condição

O elemento `Condition` (ou bloco `Condition`) permite especificar condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam operadores de condição, como "igual a" ou "menor que", para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica `OR`. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, você pode conceder a uma permissão de usuário para acessar um recurso somente se ele estiver marcado com seu nome de usuário. Para obter mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS IoT Analytics não fornece nenhuma chave de condição específica do serviço, mas oferece suporte ao uso de algumas chaves de condição globais. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Exemplos

Para ver exemplos de políticas AWS IoT Analytics baseadas em identidade, consulte. AWS IoT Analytics exemplos de políticas baseadas em identidade

AWS IoT Analytics políticas baseadas em recursos

AWS IoT Analytics não oferece suporte a políticas baseadas em recursos. Para visualizar um exemplo de uma página de política detalhada baseada em recursos, consulte Usar políticas baseadas em recursos para AWS Lambda no AWS Lambda Guia do desenvolvedor.

Autorização baseada em AWS IoT Analytics tags

Você pode anexar tags a AWS IoT Analytics recursos ou passar tags em uma solicitação para AWS IoT Analytics. Para controlar o acesso com base em tags, forneça informações sobre as tags no elemento de condição de uma política usando as chaves de condição `iotanalytics:ResourceTag/{key-name}`, `aws:RequestTag/{key-name}` ou `aws:TagKeys`. Para obter mais informações sobre como marcar AWS IoT Analytics recursos, consulte Como marcar seus AWS IoT Analytics recursos.

Para ver um exemplo de política baseada em identidade para limitar o acesso a um recurso com base nas tags desse recurso, consulte Visualização de AWS IoT Analytics canais com base em tags.

AWS IoT Analytics Funções do IAM

Um perfil do IAM é uma entidade dentro da sua Conta da AWS que tem permissões específicas.

Usando credenciais temporárias com AWS IoT Analytics

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. Você obtém credenciais de segurança temporárias chamando AWS Security Token Service (AWS STS) operações de API, como AssumeRoleou GetFederationToken.

AWS IoT Analytics não suporta o uso de credenciais temporárias.

Funções vinculadas a serviço

As funções alinhadas ao AWS serviço permitem que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. Os perfis vinculados a serviço aparecem na sua conta do IAM e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados a serviço.

AWS IoT Analytics não oferece suporte a funções vinculadas a serviços.

Perfis de serviço

Esse atributo permite que um serviço assuma um perfil de serviço em seu nome. O perfil permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. Os perfis de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso indica que um administrador do IAM pode alterar as permissões para essa função. Porém, fazer isso pode alterar a funcionalidade do serviço.

AWS IoT Analytics suporta funções de serviço.

Prevenção contra o ataque “Confused deputy” em todos os serviços

O problema ‘confused deputy’ é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a personificação entre serviços pode resultar no problema do ‘confused deputy’. 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, o AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição globais `aws:SourceArn` e `aws:SourceAccount` nas políticas de recursos. Isso limita as permissões que o AWS IoT Analytics concede a outro serviço para o 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.

A maneira mais eficaz de se proteger contra o problema substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o nome do recurso da Amazon (ARN) completo do recurso. Se você não souber o ARN completo do recurso ou se estiver especificando vários recursos, use a chave da condição de contexto global `aws:SourceArn` com curingas (*) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:iotanalytics::123456789012:*`.

Tópicos

- Prevenção para buckets do Amazon S3
- Prevenção com Amazon CloudWatch Logs
- Prevenção do substituto confuso para recursos AWS IoT Analytics gerenciados pelo cliente

Prevenção para buckets do Amazon S3

Se você usa o armazenamento gerenciado pelo cliente do Amazon S3 para seu datastore AWS IoT Analytics, o bucket do Amazon S3 que armazena seus dados pode estar exposto a problemas de representante confuso.

Por exemplo, Nikki Wolf usa um bucket do Amazon S3 de propriedade do cliente chamado *DOC-EXAMPLE-BUCKET*. O bucket armazena informações de um datastore AWS IoT Analytics que foi

criado na região *us-east-1*. Ela especifica uma política que permite que o responsável pelo serviço de entidade principal do AWS IoT Analytics consulte *DOC-EXAMPLE-BUCKET* em seu nome. A colega de trabalho de Nikki, Li Juan, consulta *DOC-EXAMPLE-BUCKET* de sua própria conta e cria um conjunto de dados com os resultados. Como resultado, a entidade principal AWS IoT Analytics consultou o bucket do Amazon S3 de Nikki em nome de Li, embora Li tenha executado a consulta em sua conta.

Para evitar isso, Nikki pode especificar a condição `aws:SourceAccount` ou a condição `aws:SourceArn` na política para *DOC-EXAMPLE-BUCKET*.

Especifique a condição **`aws:SourceAccount`** — o exemplo a seguir de uma política de bucket especifica que somente os recursos AWS IoT Analytics da conta de Nikki (*123456789012*) podem acessar *DOC-EXAMPLE-BUCKET*.

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
```

```
            }
        }
    ]
}
```

Especifique a condição **`aws:SourceArn`** — como alternativa, Nikki pode usar a condição `aws:SourceArn`.

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-
EXAMPLE-DATASET",
                        "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-
EXAMPLE-DATASTORE"
                    ]
                }
            }
        }
```

```
    ]
}
```

Prevenção com Amazon CloudWatch Logs

Você pode evitar o problema de substituto confuso ao monitorar com o Amazon CloudWatch Logs. A política de recursos a seguir mostra como evitar o problema de substituto confuso com:

- A chave contextual de condição global, `aws:SourceArn`
- O `aws:SourceAccount` com o ID da sua conta AWS
- O recurso do cliente associado à solicitação `sts:AssumeRole` no AWS IoT Analytics

Substitua *123456789012* pelo ID da sua conta AWS e *us-east-1* pela região da sua conta AWS IoT Analytics no exemplo a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "logs:PutLogEvents",
            "Resource": "*",
            "Condition":{
                "ArnLike":{
                    "aws:SourceArn":"arn:aws:iotanalytics:us-east-1:123456789012:*/*"
                },
                "StringEquals":{
                    "aws:SourceAccount":"123456789012"
                }
            }
    ]
}
```

Para obter mais informações sobre habilitar e configurar Amazon CloudWatch Logs, consulte. the section called “Registro e monitoramento”

Prevenção do substituto confuso para recursos AWS IoT Analytics gerenciados pelo cliente

Se você conceder permissão AWS IoT Analytics para realizar ações em seus recursos AWS IoT Analytics, os recursos poderão ficar expostos a problemas de substituto confuso. Para evitar o problema de substituto confuso, você pode limitar as permissões concedidas a AWS IoT Analytics com os seguintes exemplos de políticas de recursos.

Tópicos

- Prevenção para canais e armazenamentos de dados AWS IoT Analytics
- Prevenção do problema substituto confuso entre serviços para as regras de entrega do conteúdo do conjunto de dados AWS IoT Analytics

Prevenção para canais e armazenamentos de dados AWS IoT Analytics

Você usa perfil do IAM para controlar os recursos AWS que AWS IoT Analytics pode acessar em seu nome. Para evitar expor sua função ao problema de substituto confuso, você pode especificar a conta AWS no elemento `aws:SourceAccount` e o ARN do recurso AWS IoT Analytics no elemento `aws:SourceArn` da política de confiança que você atribui a uma função.

No exemplo a seguir, substitua *123456789012* pelo ID da sua conta AWS e *arn:aws:iotanalytics:aws-region:123456789012:channel/DOC-EXAMPLE-CHANNEL* pelo ARN de um canal AWS IoT Analytics ou armazenamento de dados.

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

```
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:channel/DOC-
EXAMPLE-CHANNEL"
        }
      }
    }
  ]
}
```

Para saber mais sobre as opções de armazenamento S3 gerenciado pelo cliente para canais e datastores, consulte `CustomerManagedChannelS3Storage` e `CustomerManagedDatastoreS3Storage` na AWS IoT AnalyticsReferência da API.

Prevenção do problema substituto confuso entre serviços para as regras de entrega do conteúdo do conjunto de dados AWS IoT Analytics

O perfil do IAM que o AWS IoT Analytics pressupõe fornecer resultados do da consulta do conjunto de dados o Amazon S3 ou para AWS IoT Events pode ser exposto a problemas de substituto confuso. Para evitar o problema de substituto confuso, especifique a conta AWS no elemento `aws:SourceAccount` e o ARN do recurso AWS IoT Analytics no elemento `aws:SourceArn` da política de confiança que você atribui à sua função.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExampleTrustPolicyDocument",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
       },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:dataset/DOC-
EXAMPLE-DATASET"
        }
      }
    }
  ]
```

```
}
```

Para obter mais detalhes sobre como configurar as regras de entrega de conteúdo do conjunto de dados, consulte a `contentDeliveryRules` na Referência de API do AWS IoT Analytics.

AWS IoT Analytics exemplos de políticas baseadas em identidade

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos do AWS IoT Analytics . 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 que exigem essas permissões.

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

Tópicos nesta página:

- Melhores práticas de política
- Usando o AWS IoT Analytics console
- Permitir que os usuários visualizem suas próprias permissões
- Acessando uma AWS IoT Analytics entrada
- Visualizando AWS IoT Analytics canais com base em tags

Melhores práticas de política

As políticas baseadas em identidade são muito eficientes. Eles determinam se alguém pode criar, acessar ou excluir AWS IoT Analytics recursos 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 a usar políticas AWS gerenciadas - Para começar a usar AWS IoT Analytics rapidamente, use políticas AWS gerenciadas para dar aos seus funcionários as permissões de que precisam. Essas políticas já estão disponíveis em sua conta e são mantidas e atualizadas pela AWS. Para obter mais informações, consulte Comece a usar permissões com políticas AWS gerenciadas no Guia do usuário do IAM.

- Conceder privilégio mínimo: ao criar políticas personalizadas, conceda apenas as permissões necessárias para executar uma tarefa. Comece com um conjunto mínimo de permissões e conceda permissões adicionais conforme necessário. Fazer isso é mais seguro do que começar com permissões que são muito lenientes e tentar restringi-las superiormente. Para obter mais informações, consulte Conceder privilégio mínimo no Guia do usuário do IAM.
- Habilitar o MFA para operações confidenciais: para reforçar a segurança, exija que os usuários usem a autenticação multifator (MFA) para acessar recursos ou operações de API sigilosos. Para obter mais informações, consulte Uso de autenticação multifator (MFA) na AWS no Guia do usuário do IAM.
- Usar condições de política para segurança adicional: na medida do possível, defina as condições sob as quais suas políticas baseadas em identidade permitem o acesso a um recurso. Por exemplo, você pode gravar condições para especificar um intervalo de endereços IP permitidos do qual a solicitação deve partir. Você também pode escrever condições para permitir somente solicitações em uma data especificada ou período ou para exigir o uso de SSL ou MFA. Para obter mais informações, consulte Elementos de política JSON do IAM: condição no Guia do usuário do IAM.

Usando o AWS IoT Analytics console

Para acessar o AWS IoT Analytics console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os AWS IoT Analytics recursos em seu Conta da AWS. 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) com essa política.

Para garantir que essas entidades ainda possam usar o AWS IoT Analytics console, anexe também a seguinte política AWS gerenciada às entidades. Para obter mais informações, consulte Adicionando Permissões a um Usuário no Guia do Usuário do IAM.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                 "iotanalytics:BatchPutMessage",
                 "iotanalytics:CancelPipelineReprocessing",
                 "iotanalytics:CreateChannel",
```

```
                "iotanalytics:CreateDataset",
                "iotanalytics:CreateDatasetContent",
                "iotanalytics:CreateDatastore",
                "iotanalytics:CreatePipeline",
                "iotanalytics:DeleteChannel",
                "iotanalytics:DeleteDataset",
                "iotanalytics:DeleteDatasetContent",
                "iotanalytics:DeleteDatastore",
                "iotanalytics:DeletePipeline",
                "iotanalytics:DescribeChannel",
                "iotanalytics:DescribeDataset",
                "iotanalytics:DescribeDatastore",
                "iotanalytics:DescribeLoggingOptions",
                "iotanalytics:DescribePipeline",
                "iotanalytics:GetDatasetContent",
                "iotanalytics:ListChannels",
                "iotanalytics:ListDatasetContents",
                "iotanalytics:ListDatasets",
                "iotanalytics:ListDatastores",
                "iotanalytics:ListPipelines",
                "iotanalytics:ListTagsForResource",
                "iotanalytics:PutLoggingOptions",
                "iotanalytics:RunPipelineActivity",
                "iotanalytics:SampleChannelData",
                "iotanalytics:StartPipelineReprocessing",
                "iotanalytics:TagResource",
                "iotanalytics:UntagResource",
                "iotanalytics:UpdateChannel",
                "iotanalytics:UpdateDataset",
                "iotanalytics:UpdateDatastore",
                "iotanalytics:UpdatePipeline"
            ],
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:channel/
${channelName}",
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/
${datasetName}",
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:datastore/
${datastoreName}",
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:pipeline/
${pipelineName}"
        }
    ]
 }
```

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente às ações que correspondem à operação da API que você está tentando executar.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permite que os usuários visualizem as políticas gerenciadas e embutidas 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": "*"
        }
```

```
    ]
}
```

Acessando uma AWS IoT Analytics entrada

Neste exemplo, você deseja conceder a um usuário o Conta da AWS acesso a um de seus AWS IoT Analytics canais,`exampleChannel`. Você também deseja permitir que o usuário adicione, atualize e exclua canais.

A política concede as permissões `iotanalytics:ListChannels, iotanalytics:DescribeChannel, iotanalytics:CreateChannel, iotanalytics:DeleteChannel, and iotanalytics:UpdateChannel` ao usuário. Para obter um exemplo de demonstração do serviço do Amazon S3 que concede permissões aos usuários e testa-os usando o console, consulte Um exemplo de demonstração: Usar políticas de usuário para controlar o acesso ao bucket.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ListChannelsInConsole",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:ListChannels"
         ],
         "Resource":"arn:aws:iotanalytics:::*"
      },
      {
         "Sid":"ViewSpecificChannelInfo",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:DescribeChannel"
         ],
         "Resource":"arn:aws:iotanalytics:::exampleChannel"
      },
      {
         "Sid":"ManageChannels",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:CreateChannel",
            "iotanalytics:DeleteChannel",
            "iotanalytics:DescribeChannel",
```

```
            "iotanalytics:ListChannels",
            "iotanalytics:UpdateChannel"
         ],
         "Resource":"arn:aws:iotanalytics:::exampleChannel/*"
      }
   ]
}
```

Visualizando AWS IoT Analytics canais com base em tags

Você pode usar condições em sua política baseada em identidade para controlar o acesso aos AWS IoT Analytics recursos com base em tags. Este exemplo mostra como você pode criar uma política que permite visualizar um `channel`. No entanto, a permissão é concedida somente se o `Owner` da tag `channel` tiver o valor do nome desse usuário. Essa política também concede as permissões necessárias para concluir essa ação no console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListChannelsInConsole",
            "Effect": "Allow",
            "Action": "iotanalytics:ListChannels",
            "Resource": "*"
        },
        {
            "Sid": "ViewChannelsIfOwner",
            "Effect": "Allow",
            "Action": "iotanalytics:ListChannels",
            "Resource": "arn:aws:iotanalytics:*:*:channel/*",
            "Condition": {
                "StringEquals": {"iotanalytics:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Você pode anexar essa política aos usuários na sua conta. Se um usuário chamado `richard-roe` tentar visualizar um AWS IoT Analytics `channel`, ele `channel` deverá ser marcado`Owner=richard-roe or owner=richard-roe`. Caso contrário, ele terá o acesso negado. A chave da tag de condição `Owner` corresponde a `Owner` e a `owner` porque os nomes de

chaves de condição não diferenciam letras maiúsculas de minúsculas. Para obter mais informações, consulte Condição de Elementos de Política JSON do IAM no Guia do Usuário do IAM.

Solução de problemas AWS IoT Analytics de identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com AWS IoT Analytics.

Tópicos

- Não estou autorizado a realizar uma ação em AWS IoT Analytics
- Não tenho autorização para executar iam:PassRole
- Quero permitir que pessoas fora da minha Conta da AWS acessem meus AWS IoT Analytics recursos

Não estou autorizado a realizar uma ação em AWS IoT Analytics

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 a você o seu nome de usuário e senha.

O erro de exemplo a seguir ocorre quando o usuário `mateojackson` tenta usar o console para visualizar detalhes sobre um `channel`, mas não tem as permissões `iotanalytics:ListChannels`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iotanalytics:``ListChannels`` on resource: ``my-example-channel``
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso `my-example-channel` usando a ação `iotanalytics:ListChannel`.

Não tenho autorização para executar **`iam:PassRole`**

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o AWS IoT Analytics.

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 fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta utilizar o console para executar uma ação no AWS IoT Analytics. 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.

Quero permitir que pessoas fora da minha Conta da AWS acessem meus AWS IoT Analytics recursos

Você pode criar uma função que os usuários de outras contas ou pessoas fora da sua organização possam usar para acessar seus recursos. Você pode 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 é AWS IoT Analytics compatível com esses recursos, consulte Como AWS IoT Analytics 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.
- Saiba como conceder acesso por meio da federação de identidades consultando Concedendo Acesso a Usuários Autenticados Externamente (Federação de Identidades) no Guia do Usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Registrar em log e monitorar no AWS IoT Analytics

A AWS fornece ferramentas que você pode usar para monitorar o AWS IoT Analytics. Você pode configurar algumas dessas ferramentas para que façam o monitoramento para você. Algumas das ferramentas exigem intervenção manual. Recomendamos que as tarefas de monitoramento sejam automatizadas ao máximo possível.

Ferramentas de monitoramento automatizadas

Você pode usar as seguintes ferramentas de monitoramento automatizadas para observar o AWS IoT e gerar relatórios quando algo estiver errado:

- Amazon CloudWatch Logs — monitore, armazene e acesse seus arquivos de log do AWS CloudTrail ou de outras fontes. Para obter mais informações, consulte O que é AWS CloudTrail Monitorar arquivos de log no Guia do usuário do Amazon CloudWatch.
- Monitoramento de log AWS CloudTrail: compartilhe arquivos de log entre contas, monitore os arquivos de log do CloudTrail em tempo real enviando-os para o CloudWatch Logs, escreva aplicações de processamento de logs em Java e confirme se os arquivos de log não foram alterados após a entrega pelo CloudTrail. Para obter mais informações, consulte Trabalhar com arquivos de log do CloudTrail no Guia do usuário do AWS CloudTrail.

Ferramentas de monitoramento manual

Outra parte importante do monitoramento do AWS IoT é o monitoramento manual dos itens que os alarmes do CloudWatch não abrangem. O AWS IoT, CloudWatch e outros painéis de console do serviço da AWS apresentam uma visão rápida do estado do seu ambiente na AWS. Recomendamos que você também verifique os arquivos de registro do AWS IoT Analytics.

- O console do AWS IoT Analytics mostra:
  - Canais
  - Pipelines
  - Armazenamentos de dados
  - Conjuntos de dados
  - Cadernos
  - Configurações
  - Saiba mais

- A página inicial do CloudWatch mostra:
  - Alertas e status atual
  - Gráficos de alertas e recursos
  - Estado de integridade do serviço

  Além disso, é possível usar o CloudWatch para fazer o seguinte:
  - Crie painéis personalizados para monitorar os serviços com os quais você se preocupa.
  - Colocar em gráfico dados de métrica para solucionar problemas e descobrir tendências
  - Pesquisar e procurar todas as métricas de recursos da AWS
  - Criar e editar alertas para ser notificado sobre problemas

Monitoramento com o Amazon CloudWatch Logs

AWS IoT Analytics oferece suporte ao registro em log com o Amazon CloudWatch. Você pode habilitar e configurar o registro em log do Amazon CloudWatch para AWS IoT Analytics usando a operação `PutLoggingOptions` da API. Esta seção descreve como você pode usar `PutLoggingOptions` com AWS Identity and Access Management (IAM) para configurar e habilitar o registro em log no Amazon CloudWatch para AWS IoT Analytics.

Para obter mais informações sobre o CloudWatch Logs, consulte o Guia do usuário do Amazon CloudWatch Logs Para obter mais informações sobre o IAM da AWS, consulte o Guia do usuário AWS Identity and Access Management.

> **Note**
>
> Antes de habilitar o registro em log do AWS IoT Analytics, entenda as permissões de acesso do CloudWatch Logs. Os usuários com acesso ao CloudWatch Logs podem ver suas informações de depuração. Para obter mais informações, consulte Autenticação e controle de acesso para o Amazon CloudWatch Logs.

Criar um perfil do IAM para ativar o registro em log

Para criar um perfil do IAM para ativar o registro em log para o Amazon CloudWatch

1. Use o AWSconsole do IAM ou o seguinte comando da AWSCLI do IAM, CriarPerfil, para criar um novo perfil do IAM com uma política de relacionamento de confiança (política de confiança).

A política de confiança concede a uma entidade, como o Amazon CloudWatch, permissão para assumir a função.

```
aws iam create-role --role-name exampleRoleName --assume-role-policy-document
       exampleTrustPolicy.json
```

O arquivo `exampleTrustPolicy.json` contém o conteúdo a seguir.

> **Note**
>
> Este exemplo inclui uma chave de contexto de condição global para proteger contra o problema de segurança substituto confuso. Substitua *123456789012* pela ID da sua conta AWS e *a região da AWS* pela região da AWS dos seus recursos AWS. Para obter mais informações, consulte the section called “Prevenção contra o ataque “Confused deputy” em todos os serviços”.

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

      }
   ]
}
```

Use o ARN dessa função posteriormente ao chamar o comando do AWS IoT Analytics `PutLoggingOptions`.

2. Use o IAM da AWS PutrolePolicy para anexar uma política de permissões (uma `role policy`) à função que você criou na Etapa 1.

   ```
   aws iam put-role-policy --role-name exampleRoleName --policy-name
   examplePolicyName --policy-document exampleRolePolicy.json
   ```

   O arquivo exampleRolePolicy.json contém o seguinte conteúdo:

   ```
   {
      "Version": "2012-10-17",
      "Statement": [
      {
      "Effect": "Allow",
      "Action": [
       "logs:CreateLogGroup",
       "logs:CreateLogStream"
      ],
      "Resource": [
       "arn:aws:logs:*:*:*"
      ]
      }
      ]
   }
   ```

3. Para dar permissão AWS IoT Analytics para colocar eventos de registro em log no Amazon CloudWatch, use o comando putResourcePolicy do Amazon CloudWatch.

   > **Note**
   >
   > Para ajudar a evitar o problema de segurança substituto confuso, recomendamos que você especifique `aws:SourceArn` em sua política de recursos. Isso restringe o acesso para permitir somente as solicitações provenientes de uma conta específica. Para obter mais informações sobre o problema substituto confuso, consulte the section called "Prevenção contra o ataque "Confused deputy" em todos os serviços".

```
aws logs put-resource-policy --policy-in-json
exampleResourcePolicy.json
```

O arquivo `exampleResourcePolicy.json` contém a seguinte política de recursos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "logs:PutLogEvents",
            "Resource": "*",
            "Condition":{
                "ArnLike":{
                    "aws:SourceArn":"arn:aws:iotanalytics:us-east-1:123456789012:*/
*"
                },
                "StringEquals":{
                    "aws:SourceAccount":"123456789012"
                }
            }
    ]
}
```

Configurar e habilitar o registro em log

Use o comando `PutLoggingOptions` para configurar e habilitar o registro em log do Amazon CloudWatch para AWS IoT Analytics. O `roleArn` no campo `loggingOptions` deve ser o ARN da função que você criou na seção anterior. Você também pode usar o comando `DecribeLoggingOptions` para verificar as configurações das opções de registro em log.

PutLoggingOptions

Define ou atualiza as opções de registro em log do AWS IoT Analytics. Se você atualizar o valor de qualquer campo `loggingOptions`, levará até um minuto para ver a mudança entrar em vigor. Além disso, se você alterar a política anexada à função especificada no campo `roleArn` (por exemplo,

para corrigir uma política inválida) levará até 5 minutos para que a mudança entre em vigor. Para obter mais informações, consulte `PutLoggingOptions`.

DescribeLoggingOptions

Recupera as configurações atuais das opções de registro em log do AWS IoT Analytics. Para obter mais informações, consulte `DescribeLoggingOptions`.

Namespaces, métricas e dimensões

O AWS IoT Analytics coloca as seguintes métricas no repositório do Amazon CloudWatch:

Namespace
AWS/IoTAnalytics

Métrica	Descrição
ActionExecution	O número de ações executadas.
ActionExecutionThrottled	O número de ações que são limitadas.
ActivityExecutionError	O número de erros gerados ao executar a atividade do pipeline.
IncomingMessages	O número de mensagens recebidas no canal.
PipelineConcurrentExecutionCount	O número de atividades do pipeline, que foram executadas simultaneamente.

Dimensão	Descrição
ActionType	O tipo de ação que está sendo monitorado.
ChannelName	O nome do canal que está sendo monitorado.
DatasetName	O nome do conjunto de dados que está sendo monitorado.

Dimensão	Descrição
DatastoreName	O nome do datastore que está sendo monitorado.
PipelineActivityName	O nome da atividade do pipeline que está sendo monitorada.
PipelineActivityType	O tipo da atividade do pipeline que está sendo monitorada.
PipelineName	O nome do pipeline que está sendo monitorado.

Monitorar com o Amazon CloudWatch Events

AWS IoT Analytics publica automaticamente um evento no Amazon CloudWatch Events quando ocorre um erro de runtime durante uma atividade AWS Lambda. Esse evento contém uma mensagem de erro detalhada e as chaves dos objetos do Amazon Simple Storage Service (Amazon S3) que armazenam as mensagens de canal não processadas. Você pode usar as chaves do Amazon S3 para reprocessar as mensagens do canal não processadas. Para obter mais informações, consulte Reprocessamento de mensagens do canal, a API StartPipelineReprocessing na AWS IoT AnalyticsReferência de API e O que é Amazon CloudWatch Events no Guia do usuário do Amazon CloudWatch Events.

Você também pode configurar destinos que permitam que o Amazon CloudWatch Events envie notificações ou realize outras ações. Por exemplo, você pode enviar a notificação para uma fila do Amazon Simple Queue Service (Amazon SQS) e depois invocar a API `StartReprocessingMessage` para processar as mensagens do canal salvas nos objetos do Amazon S3. O Amazon CloudWatch Events oferece suporte a vários tipos de destinos, como os seguintes:

- Amazon Kinesis Streams
- Funções do AWS Lambda
- Amazon Simple Notification Service (Amazon SNS) topics
- Filas do Amazon Simple Queue Service (Amazon SQS)

Para a lista de destinos compatíveis, consulte Destinos do Amazon EventBridge no Guia do usuário do Amazon EventBridge.

Seus recursos do CloudWatch Events e os destinos associados devem estar na região da AWS em que você criou seus recursos AWS IoT Analytics. Para obter mais informações, consulte Endpoints e cotas do serviço na Referência geral da AWS.

A notificação enviada ao Amazon CloudWatch Events sobre erros de runtime na atividade AWS Lambda usa o seguinte formato:

```
{
    "version": "version-id",
    "id": "event-id",
    "detail-type": "IoT Analytics Pipeline Failure Notification",
    "source": "aws.iotanalytics",
    "account": "aws-account",
    "time": "timestamp",
    "region": "aws-region",
    "resources": [
        "pipeline-arn"
    ],
    "detail": {
        "event-detail-version": "1.0",
        "pipeline-name": "pipeline-name",
        "error-code": "LAMBDA_FAILURE",
        "message": "error-message",
        "channel-messages": {
            "s3paths": [
                "s3-keys"
            ]
        },
        "activity-name": "lambda-activity-name",
        "lambda-function-arn": "lambda-function-arn"
    }
}
```

Exemplo de notificação:

```
{
    "version": "0",
    "id": "204e672e-ef12-09af-4cfd-de3b53673ec6",
    "detail-type": "IoT Analytics Pipeline Failure Notification",
```

```
    "source": "aws.iotanalytics",
    "account": "123456789012",
    "time": "2020-10-15T23:47:02Z",
    "region": "ap-southeast-2",
    "resources": [
        "arn:aws:iotanalytics:ap-southeast-2:123456789012:pipeline/
test_pipeline_failure"
    ],
    "detail": {
        "event-detail-version": "1.0",
        "pipeline-name": "test_pipeline_failure",
        "error-code": "LAMBDA_FAILURE",
        "message": "Temp unavaliable",
        "channel-messages": {
        "s3paths": [
            "test_pipeline_failure/channel/cmr_channel/__dt=2020-10-15
 00:00:00/1602805530000_1602805560000_123456789012_cmr_channel_0_257.0.json.gz"
        ]
    },
    "activity-name": "LambdaActivity_33",
    "lambda-function-arn": "arn:aws:lambda:ap-
southeast-2:123456789012:function:lambda_activity"
    }
}
```

Receber notificações de dados atrasadas por meio do Amazon CloudWatch Events

Quando você cria conteúdo do conjunto de dados usando dados da mensagem de um período especificado, alguns dados da mensagem ainda podem não chegar a tempo para serem processados. Para permitir um atraso, você pode especificar um deslocamento de `deltaTime` para o `QueryFilter` ao criar um conjunto de dados aplicando uma `queryAction` (uma consulta SQL). AWS IoT Analytics ainda processa os dados que chegam dentro do tempo delta, e o conteúdo do conjunto de dados tem um intervalo de tempo. O atributo de notificação atrasada de dados permite ao AWS IoT Analytics enviar notificações por meio do Amazon CloudWatch Events quando os dados chegam após o horário delta.

Você pode usar o console AWS IoT Analytics, a API, AWS Command Line Interface(AWS CLI) ou o SDK da AWS para especificar regras de dados atrasados para um conjunto de dados.

Na API AWS IoT Analytics, o objeto `LateDataRuleConfiguration` representa as configurações da regra de dados atrasados de um conjunto de dados. Esse objeto faz parte do objeto `Dataset` associado a `CreateDataset` e às operações da API `UpdateDataset`.

Parâmetros

Ao criar uma regra de dados atrasados para um conjunto de dados com AWS IoT Analytics, é necessário especificar as seguintes informações:

`ruleConfiguration` (**`LateDataRuleConfiguration`**)

Uma estrutura que contém as informações de configuração de uma regra de dados atrasada.

`deltaTimeSessionWindowConfiguration`

Uma estrutura que contém as informações de configuração de uma janela de sessão de tempo delta.

DeltaTime especifica um intervalo de tempo. Você pode usar `DeltaTime` para criar conteúdo de conjunto de dados com dados que chegaram ao armazenamento de dados desde a última execução. Para obter um exemplo de `DeltaTime`, consulte Criação de um conjunto de dados SQL com uma janela delta (CLI).

`timeoutInMinutes`

Um intervalo de tempo. Você pode usar `timeoutInMinutes` de forma que o AWS IoT Analytics possa agrupar notificações de dados atrasadas que foram geradas desde a última execução. O AWS IoT Analytics envia um lote de notificações para o CloudWatch Events ao mesmo tempo.

Tipo: inteiro

Intervalo válido: 1-60

`ruleName`

O nome da regra de dados atrasados.

Tipo: sequência

> **Important**
>
> Para especificar `lateDataRules`, o conjunto de dados deve usar um filtro `DeltaTime`.

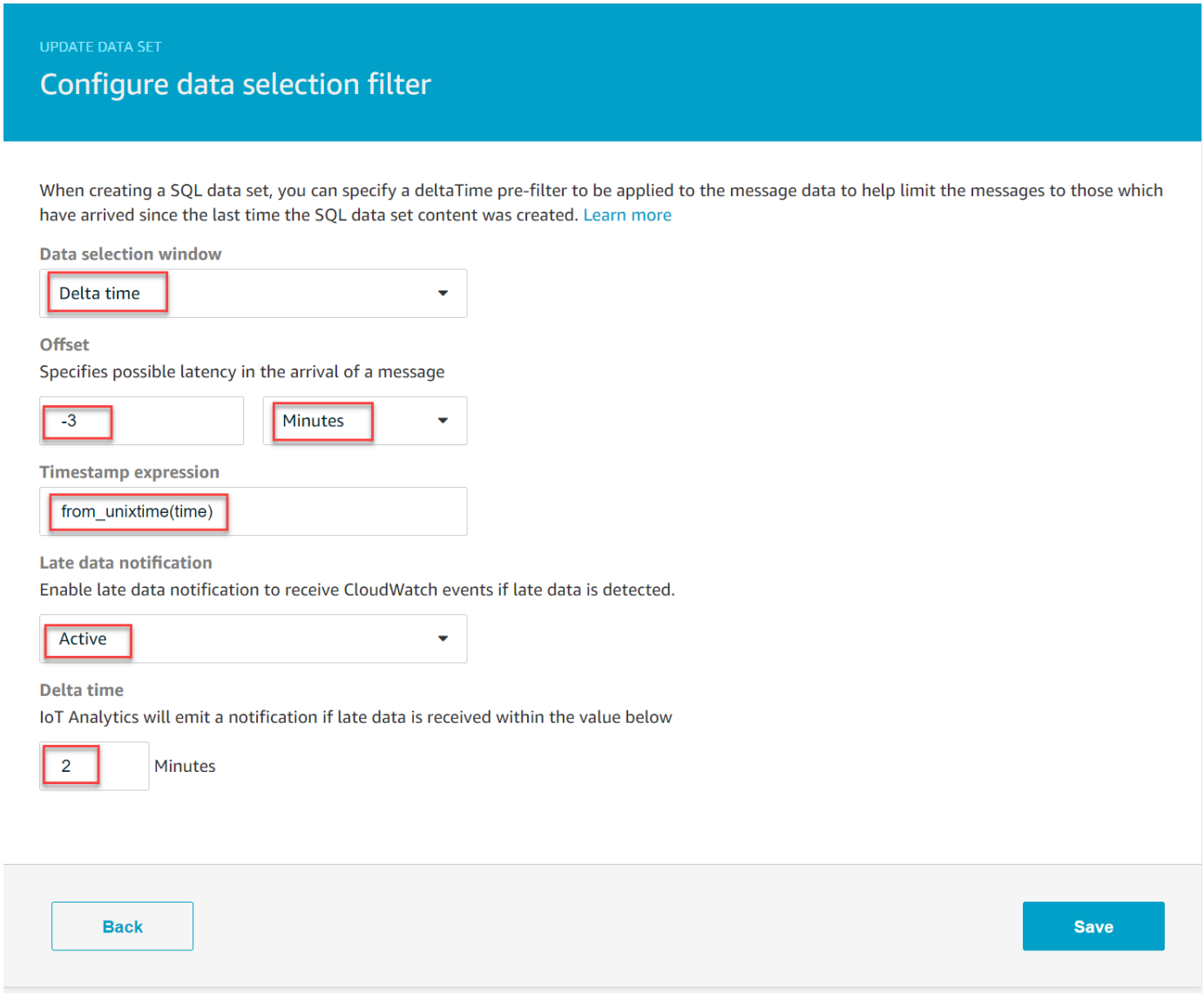
Configurar regras de dados atrasados (console)

O procedimento a seguir mostra como configurar a regra de dados atrasados de um conjunto de dados no console AWS IoT Analytics.

Para configurar regras de dados atrasados

1. Faça login no console do AWS IoT Analytics.
2. No painel de navegação, escolha Conjunto de dados.
3. Em Conjuntos de dados, escolha o conjunto de dados de destino.
4. No painel de navegação, escolha Detalhes.
5. Na seção Janela delta, escolha Editar.
6. Em Configurar filtro de seleção de dados, faça o seguinte:
   a. Em Janela de seleção de dados, escolha Hora delta.
   b. Em Deslocamento, insira um período de tempo e escolha uma unidade.
   c. Em Expressão de timestamp, insira uma expressão. Pode ser o nome de um campo de timestamp ou uma expressão SQL que pode derivar a hora, como `from_unixtime(time)`.

      Para obter mais informações sobre como escrever uma expressão timestamp, consulte Funções de data e hora e operadores, na Documentação do Presto 0.172.
   d. Para Notificação de dados atrasada, escolha Ativo.
   e. Em Hora delta, insira um número inteiro. O intervalo válido é 1-60.
   f. Escolha Save (Salvar).



Configurar regras de dados atrasada (CLI)

Na API AWS IoT Analytics, o objeto `LateDataRuleConfiguration` representa as configurações da regra de dados atrasados de um conjunto de dados. Esse objeto faz parte do objeto `Dataset` associado a `CreateDataset` e `UpdateDataset`. Você pode usar a API, AWS CLI ou AWSSDK para especificar regras de dados atrasados para um conjunto de dados. O exemplo a seguir usa a AWS CLI.

Para criar o conjunto de dados com regras de dados atrasados especificadas, execute o comando a seguir. O comando a seguir pressupõe que o arquivo `dataset.json` esteja no diretório atual.

> **Note**
>
> Você pode usar a API UpdateDataset para atualizar um conjunto de dados existente.

```
aws iotanalytics create-dataset --cli-input-json file://dataset.json
```

O arquivo `dataset.json` deve conter o seguinte:

- Substitua *`demo_dataset`* pelo nome do conjunto de dados de destino.
- Substitua *`demo_datastore`* pelo nome do datastore de destino.
- Substitua *`from_unixtime(time)`* pelo nome de um campo de timestamp ou uma expressão SQL que possa derivar a hora.

  Para obter mais informações sobre como escrever uma expressão timestamp, consulte Funções de data e hora e operadores, na Documentação do Presto 0.172.
- Substitua *`o tempo limite`* por um número inteiro entre 1-60.
- Substitua *`demo_rule`* por qualquer nome.

```
{
    "datasetName": "demo_dataset",
    "actions": [
        {
            "actionName": "myDatasetAction",
            "queryAction": {
                "filters": [
                    {
                        "deltaTime": {
                            "offsetSeconds": -180,
                            "timeExpression": "from_unixtime(time)"
                        }
                    }
                ],
                "sqlQuery": "SELECT * FROM demo_datastore"
            }
        }
    ],
    "retentionPeriod": {
```

```
        "unlimited": false,
        "numberOfDays": 90
    },
    "lateDataRules": [
        {
            "ruleConfiguration": {
                "deltaTimeSessionWindowConfiguration": {
                    "timeoutInMinutes": timeout
                }
            },
            "ruleName": "demo_rule"
        }
    ]
}
```

Assinando para receber notificações de dados atrasados

Você pode criar regras no CloudWatch Events que definem como processar notificações de dados atrasadas enviadas de AWS IoT Analytics. Quando o CloudWatch Events recebe as notificações, ele invoca as ações do destino especificadas nas suas regras.

Pré-requisitos para criar regras do CloudWatch Events

Antes de criar uma regra do CloudWatch Events para o AWS IoT Analytics, é necessário:

- Familiarize-se com os eventos, as regras e os destinos no CloudWatch Events.
- Crie e configure os destinos invocados por suas regras do CloudWatch Events. As regras podem invocar muitos tipos de destinos, como:
  - Amazon Kinesis Streams
  - Funções do AWS Lambda
  - Amazon Simple Notification Service (Amazon SNS) topics
  - Filas do Amazon Simple Queue Service (Amazon SQS)

  Sua regra do CloudWatch Events e os destinos associados devem estar na região da AWS em que você criou seus recursos AWS IoT Analytics. Para obter mais informações, consulte Endpoints e cotas do serviço na Referência geral da AWS.

Para obter mais informações, consulte O que é o CloudWatch Events? e Conceitos básicos do Amazon CloudWatch Events no Guia do usuário do Amazon CloudWatch Events.

Evento de notificação de dados atrasados

O evento para notificações de dados atrasados usa o formato a seguir.

```
{
 "version": "0",
 "id": "7f51dfa7-ffef-97a5-c625-abddbac5eadd",
 "detail-type": "IoT Analytics Dataset Lifecycle Notification",
 "source": "aws.iotanalytics",
 "account": "123456789012",
 "time": "2020-05-14T02:38:46Z",
 "region": "us-east-2",
 "resources": ["arn:aws:iotanalytics:us-east-2:123456789012:dataset/demo_dataset"],
 "detail": {
  "event-detail-version": "1.0",
  "dataset-name": "demo_dataset",
  "late-data-rule-name": "demo_rule",
  "version-ids": ["78244852-8737-4650-aa4d-3071a01338fa"],
  "message": null
 }
}
```

Criar uma regra do CloudWatch Events para receber notificações de dados atrasados

O procedimento a seguir mostra como criar uma regra que envia notificações de dados AWS IoT Analytics atrasados para uma fila do Amazon SQS.

Para criar uma regra do CloudWatch Events

1. Faça login no console do Amazon CloudWatch.
2. No painel de navegação, em Events (Eventos), escolha Rules (Regras).
3. Na página Regras, selecione Criar uma regra.
4. Em Fonte do evento, selecione Padrão do evento.
5. Na seção Construir padrão de eventos para corresponder a eventos por serviço, faça o seguinte:
   a. Em Nome do serviço, escolha IoT Analytics
   b. Em Tipo de evento, escolha Notificação do ciclo de vida do conjunto de dados do IoT Analytics.
   c. Escolha nome(s) específicos do conjunto de dados e, em seguida, insira o nome do conjunto de dados de destino.

6. Em Destinos, escolha Adicionar destino*.
7. Selecione Fila do SQS e faça o seguinte:
   - Em Fila*, escolha a fila de destino.
8. Escolha Configure details (Configurar detalhes).
9. Na página Etapa 2: Configurar detalhes da regra insira um nome e uma descrição.
10. Escolha Create rule (Criar regra).

Registrar em log chamadas de API do AWS IoT Analytics com o AWS CloudTrail

O AWS IoT Analytics é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, uma função ou um serviço da AWS no AWS IoT Analytics. O CloudTrail captura todas as chamadas de API para o AWS IoT Analytics como eventos, incluindo as chamadas do console do AWS IoT Analytics e de chamadas de código para APIs do AWS IoT Analytics. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o AWS IoT Analytics. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o AWS IoT Analytics, endereço IP no qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita, além de detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte o Guia do usuário do AWS CloudTrail.

Informações do AWS IoT Analytics no AWS CloudTrail

O CloudTrail é habilitado em sua conta da AWS quando ela é criada. Quando ocorre uma atividade no AWS IoT Analytics, essa atividade é registrada em um evento do CloudTrail com outros eventos de produtos da AWS em Event history (Histórico de eventos). Você pode visualizar, pesquisar e baixar os eventos recentes em sua conta da AWS. Para obter mais informações, consulte Visualizar eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo de eventos na conta da AWS, incluindo eventos do AWS IoT Analytics, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões da . A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível

configurar outros serviços da AWS para analisar mais ainda mais e agir com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- Serviços e integrações compatíveis com o CloudTrail
- Configurar notificações do Amazon SNS para o CloudTrail
- Receiving CloudTrail log files from multiple regions e Receiving CloudTrail log files from multiple accounts

O AWS IoT Analytics é compatível com as seguintes ações como eventos nos arquivos de log do CloudTrail:

- CancelPipelineReprocessing
- CreateChannel
- CreateDataset
- CreateDatasetContent
- CreateDatastore
- CreatePipeline
- DeleteChannel
- DeleteDataset
- DeleteDatasetContent
- DeleteDatastore
- DeletePipeline
- DescribeChannel
- DescribeDataset
- DescribeDatastore
- DescribeLoggingOptions
- DescribePipeline
- GetDatasetContent
- ListChannels
- ListDatasets
- ListDatastores

- ListPipelines
- PutLoggingOptions
- RunPipelineActivity
- SampleChannelData
- StartPipelineReprocessing
- UpdateChannel
- UpdateDataset
- UpdateDatastore
- UpdatePipeline

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário da raiz ou do AWS Identity and Access Management.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail.

Noções básicas sobre entradas de arquivos de log do AWS IoT Analytics

Uma trilha é uma configuração que permite a entrega de eventos como registros de log a um bucket do S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `CreateChannel`.

```
{
"eventVersion": "1.05",
"userIdentity": {
```

```
"type": "AssumedRole",
"principalId": "ABCDE12345FGHIJ67890B:AnalyticsChannelTestFunction",
"arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/
AnalyticsChannelTestFunction",
"accountId": "123456789012",
"accessKeyId": "ABCDE12345FGHIJ67890B",
"sessionContext": {
"attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2018-02-14T23:43:12Z"
},
"sessionIssuer": {
 "type": "Role",
 "principalId": "ABCDE12345FGHIJ67890B",
 "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
 "accountId": "123456789012",
 "userName": "AnalyticsRole"
}
}
},
"eventTime": "2018-02-14T23:55:14Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateChannel",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
"channelName": "channel_channeltest"
},
"responseElements": {
"retentionPeriod": {
"unlimited": true
},
"channelName": "channel_channeltest",
"channelArn": "arn:aws:iotanalytics:us-east-1:123456789012:channel/channel_channeltest"
},
"requestID": "7f871429-11e2-11e8-9eee-0781b5c0ac59",
"eventID": "17885899-6977-41be-a6a0-74bb95a78294",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `CreateDataset`.


```
{
"eventVersion": "1.05",
"userIdentity": {
"type": "AssumedRole",
"principalId": "ABCDE12345FGHIJ67890B:AnalyticsDatasetTestFunction",
"arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/
AnalyticsDatasetTestFunction",
"accountId": "123456789012",
"accessKeyId": "ABCDE12345FGHIJ67890B",
"sessionContext": {
"attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2018-02-14T23:41:36Z"
},
"sessionIssuer": {
 "type": "Role",
 "principalId": "ABCDE12345FGHIJ67890B",
 "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
 "accountId": "123456789012",
 "userName": "AnalyticsRole"
}
}
},
"eventTime": "2018-02-14T23:53:39Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateDataset",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
"datasetName": "dataset_datasettest"
},
"responseElements": {
"datasetArn": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/
dataset_datasettest",
"datasetName": "dataset_datasettest"
},
"requestID": "46ee8dd9-11e2-11e8-979a-6198b668c3f0",
"eventID": "5abe21f6-ee1a-48ef-afc5-c77211235303",
"eventType": "AwsApiCall",
```

```
"recipientAccountId": "123456789012"
}
```

Validação de conformidade para AWS IoT Analytics

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:

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

  > **Note**
  >
  > Nem todos Serviços da AWS são elegíveis para a HIPAA. Para obter mais informações, consulte Referência dos Serviços Qualificados pela 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 atributos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços com suporte 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.

Resiliência em AWS IoT Analytics

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, conectadas com baixa latência, throughput elevado e redes altamente redundantes. Com as zonas de disponibilidade, você pode projetar e operar aplicações e bancos de dados que executam o failover automaticamente entre as Zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenter tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte infraestrutura AWS global.

Segurança da infraestrutura em AWS IoT Analytics

Como serviço gerenciado, AWS IoT Analytics 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 pela rede. Os clientes devem ser compatíveis 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, suporta esses modos.

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

AWS IoT AnalyticsCotas

O Guia Referência geral da AWS fornece as cotas padrão do AWS IoT Analytics para uma conta da AWS. Salvo indicação em contrário, cada cota aplica-se por região da AWS. Para obter mais informações, consulte AWS IoT AnalyticsEndpoints e cotas e AWS Service Quotas no Guia Referência geral da AWS.

Para solicitar um aumento das Service Quotas, envie um caso de suporte no console da Central de suporte. Para obter mais informações, consulte Requesting a quota increase (Como solicitar um aumento de cota) no Service Quotas User Guide (Guia do usuário do Service Quotas).

Comandos AWS IoT Analytics

Leia este tópico para saber mais sobre as operações de API para AWS IoT Analytics, incluindo solicitações de exemplos, respostas e erros para os protocolos de serviços web compatíveis.

Ações AWS IoT Analytics

Você pode usar comandos de API AWS IoT Analytics para coletar, processar, armazenar e analisar seus dados de IoT. Para obter mais informações, consulte as ações que são compatíveis com AWS IoT Analytics na AWS IoT AnalyticsReferência da API.

As AWS IoT Analyticsseções na AWS CLIReferência de Comandos incluem os comandos da AWS CLI que você pode usar para administrar e manipular o AWS IoT Analytics.

Dados do AWS IoT Analytics

Você pode usar os comandos da API de dados AWS IoT Analytics para realizar atividades avançadas com AWS IoT Analytics, `channel`, `pipeline`, `datastore` e `dataset`. Para obter mais informações, consulte os tipos de dados que tesão compatíveis com os dados AWS IoT Analytics na Referência de API do AWS IoT Analytics.

Solução de problemas AWS IoT Analytics

Consulte a seção a seguir para solucionar erros e encontrar possíveis soluções para resolver problemas com AWS IoT Analytics.

Tópicos

- Como saber se minhas mensagens estão chegando no AWS IoT Analytics?
- Por que meu pipeline perde mensagens? Como posso corrigir isso?
- Por que não há dados em meu datastore?
- Por que meu conjunto de dados simplesmente mostra __dt?
- Como fazer para codificar um evento orientado pela conclusão do conjunto de dados?
- Como fazer para configurar corretamente minha instância de caderno para usar o AWS IoT Analytics?
- Por que não consigo criar cadernos em uma instância?
- Por que não estou vendo meus conjuntos de dados no Amazon QuickSight?
- Por que não vejo o botão conteinerizar em meu caderno Jupyter existente?
- Por que minha instalação do plug-in de conteinerização está falhando?
- Por que meu plug-in de conteinerização está emitindo um erro?
- Por que não vejo minhas variáveis durante a conteinerização?
- Quais variáveis posso adicionar a meu contêiner como uma entrada?
- Como faço para definir a saída de meu contêiner como uma entrada para a análise subsequente?
- Por que meu conjunto de dados de contêiner está falhando?

Como saber se minhas mensagens estão chegando no AWS IoT Analytics?

Verifique se a regra para injetar dados no canal por meio do mecanismo de regras está configurada corretamente.

```
aws iot get-topic-rule --rule-name your-rule-name
```

A resposta deve ser parecida com o seguinte:

```
{
    "ruleArn": "arn:aws:iot:us-west-2:your-account-id:rule/your-rule-name",
    "rule": {
        "awsIotSqlVersion": "2016-03-23",
        "sql": "SELECT * FROM 'iot/your-rule-name'",
        "ruleDisabled": false,
        "actions": [
            {
                "iotAnalytics": {
                    "channelArn":
 "arn:aws:iotanalytics:region:your_account_id:channel/your-channel-name"
                }
            }
        ],
        "ruleName": "your-rule-name"
    }
}
```

Certifique-se de que o nome da região e do canal usados na regra estão corretos. Para garantir que os dados atinjam o mecanismo de regras e a regra está sendo executada corretamente, é possível adicionar um novo destino para armazenar mensagens recebidas no bucket do Amazon S3 temporariamente.

Por que meu pipeline perde mensagens? Como posso corrigir isso?

- Uma atividade recebeu uma entrada JSON inválida:

  Todas as atividades, exceto as atividades do Lambda, exigem especificamente uma string JSON válida como entrada. Se o JSON recebido por uma atividade for inválido, a mensagem é descartada e não faz seu caminho para o datastore. Verifique se você está consumindo mensagens JSON válidas para o serviço. Em caso de entrada de binário, certifique-se de que a primeira atividade no pipeline é uma atividade do Lambda que converte dados binários em JSON válido antes de transmiti-lo para a próxima atividade ou armazená-lo no datastore. Para obter mais informações, consulte Exemplo 2 da função do Lambda.

- Uma função do Lambda invocada por uma atividade do Lambda tem permissões insuficientes:

Certifique-se de que cada função do Lambda em uma atividade do Lambda tem permissão para ser invocada no serviço do AWS IoT Analytics. Os seguintes comando da AWS CLI podem ser usados para conceder permissão:

```
aws lambda add-permission --function-name <name> --region <region> --statement-id
 <id> --principal iotanalytics.amazonaws.com --action lambda:InvokeFunction
```

- Um filtro ou atividade removeAttribute é definida incorretamente:

  Certifique-se de que as definições de qualquer atividade do `filter` ou `removeAttribute` estão corretas. Se você filtrar uma mensagem ou remover todos os atributos de uma mensagem, essa mensagem não é adicionada ao datastore.

Por que não há dados em meu datastore?

- Existe um atraso entre ingestão de dados e a disponibilidade de dados:

  Pode demorar vários minutos depois de os dados serem ingeridos em um canal antes que os dados estejam disponíveis no datastore. O tempo varia com base no número de atividades do pipeline e na definição de qualquer atividade do Lambda personalizadas no pipeline.
- As mensagens estão sendo filtradas no pipeline:

  Certifique-se de que você não está soltando mensagens no pipeline. (Consulte a pergunta e resposta anteriores.)
- Sua consulta de conjunto de dados está incorreta:

  Certifique-se de que a consulta que gera o conjunto de dados do datastore está correta. Remova os filtros desnecessários da consulta para garantir que seus dados chegam ao datastore.

Por que meu conjunto de dados simplesmente mostra **`__dt`**?

- Essa coluna é adicionada automaticamente pelo serviço e contém o tempo aproximado de ingestão dos dados. Ela pode ser usada para otimizar as consultas. Se seu conjunto de dados não contiver nada além disso, consulte a pergunta e a resposta anteriores.

Como fazer para codificar um evento orientado pela conclusão do conjunto de dados?

- Será necessário configurar a sondagem com base no comando `describe-dataset` para verificar se o status do conjunto de dados com determinado timestamp é BEM-SUCEDIDO.

Como fazer para configurar corretamente minha instância de caderno para usar o AWS IoT Analytics?

Siga estas etapas para garantir que a função do IAM que você está usando para criar a instância do bloco de anotações tem as permissões necessárias:

1. Vá para o console do SageMaker e crie uma instância de caderno.
2. Preencha os detalhes e selecione create a new role (criar uma nova função). Anote o Role ARN (ARN da função).
3. Crie a instância de bloco de anotações. Isto também cria uma função que o SageMaker pode usar.
4. Vá para o console do IAM e modifique a função recém-criada do Sagemaker. Quando você abrir essa função, ela deve ter uma política gerenciada.
5. Clique em adicionar política em linha, escolha IoTAnalytics como o serviço e, em permissão de leitura, selecione GetDatasetContent.
6. Analise a política, adicione um nome para a política e, em seguida, create (criar). Agora a função recém-criada tem política de permissão para ler um conjunto de dados do AWS IoT Analytics.
7. Vá para o console do AWS IoT Analytics e crie cadernos na instância de caderno.
8. Aguarde até que a instância do blocos de anotações esteja no estado "In Service" (Em serviço).
9. Escolha criar cadernos e selecione a instância de caderno que você criou. Isto cria um caderno Jupyter com o modelo selecionado que pode acessar seus conjuntos de dados.

Por que não consigo criar cadernos em uma instância?

- Certifique-se de criar uma instância de blocos de anotações correta com a política do IAM. (Siga as etapas na pergunta anterior.)

- Certifique-se de que a instância de blocos de anotações está no estado "Em serviço". Ao criar uma instância, ela é iniciada em um estado “Pendente”. Geralmente, demora aproximadamente cinco minutos para que ela entre no estado "In Service" (Em serviço). Se a instância de cadernos entrar no estado “Falha” após cinco minutos, verifique as permissões novamente.

Por que não estou vendo meus conjuntos de dados no Amazon QuickSight?

O Amazon QuickSight pode precisar de permissão para ler o conteúdo do seu conjunto de dados do AWS IoT Analytics. Para dar permissão, siga estas etapas:

1. Escolha o nome da sua conta no canto superior direito do Amazon QuickSight e escolha Gerenciar QuickSight.
2. No painel de navegação esquerdo, escolha Segurança e permissões. Em Acesso do QuickSight aos serviços da AWS, verifique se o acesso foi concedido a AWS IoT Analytics.
   a. Se AWS IoT Analytics não tiver acesso, escolha Adicionar ou remover.
   b. Escolha a caixa ao lado AWS IoT Analytics e selecione Atualizar. Isso concede permissões de leitura do Amazon QuickSight para o conteúdo de seus conjuntos de dados.
3. Tente novamente para visualizar seus dados.

Certifique-se de escolher a mesma região da AWS tanto para o AWS IoT Analytics quanto para o Amazon QuickSight. Caso contrário, você poderá ter problemas para acessar os recursos AWS. Para ver a lista de regiões compatíveis, consulte AWS IoT Analyticsendpoints e cotas e endpoints e cotas do Amazon QuickSight no Referência geral da Amazon Web Services.

Por que não vejo o botão conteinerizar em meu caderno Jupyter existente?

- Isso é causado por um plug-in ausente de conteinerização da AWS IoT Analytics. Se você criou sua instância de caderno do SageMaker antes de 23 de agosto de 2018, será necessário instalar o plug-in manualmente, seguindo as instruções em Conteinerização de um caderno.
- Se você não vir o botão conteinerizar depois de criar a instância de caderno do SageMaker no console do AWS IoT Analytics ou ao instalá-la manualmente, entre em contato com o Suporte técnico do AWS IoT Analytics.

Por que minha instalação do plug-in de conteinerização está falhando?

- Geralmente, a instalação do plug-in falha devido à ausência de permissões na instância de notebook do SageMaker. Para obter as permissões necessárias para a instância de notebook, consulte Permissões e adicione as permissões necessárias para a função de instância de notebook. Se o problema persistir, crie uma nova instância do caderno no console do AWS IoT Analytics.
- Você pode ignorar a mensagem a seguir no log se ela aparecer durante a instalação do plug-in: “Para inicializar essa extensão no navegador sempre que o upload do caderno (ou de outro aplicativo) é feito”.

Por que meu plug-in de conteinerização está emitindo um erro?

- A conteinerização pode falhar e gerar erros por vários motivos. Verifique se você está usando o kernel correto antes de conteinerizar seu notebook. Os kernels conteinerizados começam com o prefixo "Containerized".
- Como o plug-in cria e salva uma imagem de docker em um repositório do ECR, verifique se sua função de instância de notebook tem permissões suficientes para ler, listar e criar repositórios do ECR. Para obter as permissões necessárias para a instância de notebook, consulte Permissões e adicione as permissões necessárias para a função de instância de notebook.
- Além disso, verifique se o nome do repositório está em conformidade com os requisitos do ECR. Os nomes de repositório do ECR devem começar com uma letra e podem conter apenas letras minúsculas, números, hífens, sublinhados e barras.
- Se o processo de conteinerização falhar com o erro: "Esta instância tem espaço livre insuficiente para executar a conteinerização" tente usar uma instância maior para resolver o problema.
- Se você vir erros de conexão ou um erro de criação de imagem, tente novamente. Se o problema persistir, reinicie a instância e instale a versão mais recente do plug-in.

Por que não vejo minhas variáveis durante a conteinerização?

- O plug-in de conteinerização do AWS IoT Analytics reconhece automaticamente todas as variáveis em seu caderno depois de executar o caderno com o kernel “conteinerizado”. Use um dos kernels conteinerizados para executar o notebook e, em seguida, execute a conteinerização.

Quais variáveis posso adicionar a meu contêiner como uma entrada?

- Você pode adicionar qualquer variável cujo valor queira modificar durante o tempo de execução como uma entrada para o contêiner. Isso permite executar o mesmo contêiner com diferentes parâmetros que precisam ser fornecidos no momento da criação do conjuntos de dados. O plug-in Jupyter de conteinerização do AWS IoT Analytics simplifica esse processo reconhecendo automaticamente as variáveis no bloco de anotações e disponibilizando-as como parte do processo de conteinerização.

Como faço para definir a saída de meu contêiner como uma entrada para a análise subsequente?

- Um local específico no S3 onde os artefatos executados podem ser armazenados é criado para cada execução de seu conjunto de dados de contêiner. Para acessar esse local de saída, crie uma variável com o tipo `outputFileUriValue` em seu conjunto de dados de contêiner. O valor dessa variável deve ser um caminho do S3 que é usado para armazenar arquivos de saída adicionais. Para acessar esses artefatos salvos em execuções subsequentes, você pode usar a API `getDatasetContent` e escolher o arquivo de saída apropriado para a execução subsequente.

Por que meu conjunto de dados de contêiner está falhando?

- Verifique se você está passando a `executionRole` correta para o conjunto de dados de contêiner. A política de confiança da `executionRole` deve incluir `iotanalytics.amazonaws.com` e `sagemaker.amazonaws.com`.
- Se você vir `AlgorithmError` como o motivo da falha, tente depurar o código do contêiner manualmente. Isso acontece quando há um bug no código do contêiner ou quando a função de execução não tem permissão para executar o contêiner. Se você tiver conteinerizado usando o plug-in Jupyter do AWS IoT Analytics, crie uma instância de caderno do SageMaker com a mesma função de executionRole do containerDataset e tente executar o caderno manualmente. Se o contêiner tiver sido criado fora do plug-in Jupyter, tente executar o código manualmente e limitar a permissão para a executionRole.

Histórico do documento

A tabela a seguir descreve alterações importantes no Guia do usuário AWS IoT Analytics após 3 de novembro de 2020. Para obter mais informações sobre as atualizações desta documentação, você pode se tornar assinante de um feed RSS.

Alteração	Descrição	Data
Lançamento regional	O AWS IoT Analytics está disponível agora na região da Ásia-Pacífico (Mumbai).	18 de agosto de 2021
Consulta com `JOIN`	Essa atualização permite que você use `JOIN` para consultar um conjunto de dados AWS IoT Analytics.	27 de julho de 2021
Integração com AWS IoT SiteWise	Agora você pode usar AWS IoT Analytics para consultar dados AWS IoT SiteWise.	27 de julho de 2021
Partições personalizadas	AWS IoT Analytics agora geralmente suporta o particion amento de seus dados de acordo com atributos de mensagem ou atributos adicionados por meio de atividades de pipeline.	14 de junho de 2021
Reprocessamento de mensagens do canal	Essa atualização permite que você reprocesse os dados do canal nos objetos do Amazon S3 especificados.	15 de dezembro de 2020
Esquema do Parquet	Datastores AWS IoT Analytics agora suportam o formato de arquivo Parquet.	15 de dezembro de 2020

Monitoramento com CloudWatch Events	AWS IoT Analytics publica automaticamente um evento no Amazon CloudWatch Events quando ocorre um erro de runtime durante uma atividade AWS Lambda.	15 de dezembro de 2020
Notificação de dados atrasada	Você pode usar esse atributo para receber notificações por meio do Amazon CloudWatch Events quando os dados atrasados chegam.	9 de novembro de 2020
Lançamento regional	Lançado AWS IoT Analytics na China (Pequim).	4 de novembro de 2020

Atualizações anteriores

A tabela a seguir descreve alterações importantes no Guia do usuário AWS IoT Analytics antes de 4 de novembro de 2020.

Alteração	Descrição	Data
Lançamento regional	Lançado AWS IoT Analytics na região Ásia-Pacífico (Sydney).	16 de julho de 2020
Atualização	Documentação reorganizada.	7 de maio de 2020

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.